

# QL Today

Volume 2  
Issue 3  
Sept./October  
1997

ISSN 1432-5454

The Magazine about QL, QDOS,  
Sinclair Computers, SMSQ...

## THE CONVENTION

QLers?

QSA?

? SQL?

QDOSSERS?

?

QUANTICS?

?

QLIVERS?

## Build your own QL Peripherals

Stuart Honeyball starts a series  
of hardware fundamentals

# QL Today

ISSN 1432-5454

## German office & Publisher:

Jochen Merz Software  
Im stillen Winkel 12  
47169 Duisburg  
Germany  
Tel. +49 203 502011  
Fax +49 203 502012  
Box1 +49 203 502013  
Box2 +49 203 502014

## English office:

Q Branch  
P.O. Box 7  
East Sussex, United Kingdom BN41 2ND  
Tel. +44 1273 386030  
Fax +44 1273 381577

## Editor:

Dilwyn Jones  
41 Bro Emrys  
Tal-Y-Bont, Bangor  
Gwynedd, United Kingdom LL57 3YT  
Tel. +44 1248 354023  
Fax +44 1248 354023

**QL Today** is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Subscription rates are as follows:

Germany	DM 70,-
England	DM 60,- or £25
Rest of the World	DM 70,- or £30

Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque. Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) in ASCII, Quill or text87 format. Pictures may be in \_SCR format, we can also handle GIF or TIF. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising DEADLINES are as follows:

Issue 1: 15 April	Issue 2: 15 June
Issue 3: 15 August	Issue 4: 15 October
Issue 5: 15 December	Issue 6: 15 February

**QL Today** reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine is copyrighted and all material published remains the property of **QL Today** unless otherwise specified. Written permission from **QL Today** is required before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

# Contents

- 3 Editorial
- 4 News
- 6 Arborescent Sorting - Stephen Poole
- 9 Display Code - Dilwyn Jones
- 18 QPC - My Experiences - Norman Dunbar
- 22 ProWesS Delivers - Part 2 - James Hunkins
- 31 IBMDISK Ver 2.00 - Darren D. Branagh
- 34 Multi-Processing, what is it good for? - Part 2 - Nasta (Zeljko Nastasic)
- 36 Snippet's Corner - Part 6 - M. Knight
- 42 Logical Fundamentals - Part 1 - Stuart Honeyball
- 44 Label Typewriter - Dilwyn Jones
- 45 London QL Traders Show - Dilwyn Jones
- 48 Are you sitting comfortably? (Part 1) - Geoff Wicks
- 51 Getting into the States - Roy Wood
- 52 Bugs 'n Fixes
- 55 Letter Box
- 58 Mailbox & eMail Shortcuts - Jochen Merz

# Advertisers

in alphabetical order

Jochen Merz Software	33, 59
PROGS - Van der Auwera	37
QBOX USA	53
QBranch	24-25
Quanta	17
Qubbesoft	41
W.N. Richardson (EEC)	49
Steve Johnson	57
TF Services	7

# Oops...cover disk bug!

We missed a bug in the startup routines of the cover disk supplied with the last issue. Ivan M. Dixon reported that an error 'at line 1890 bad line' was reported. The fix is simple (thanks Ivan):

Starting from line 1890 of the BOOT program, change the FOR loop control variable 'num%' to a non-integer variable such as 'num'. This applies to lines 1890, 1900 and 1910.

■

# Editorial

Dilwyn Jones

As avid QL users, we all (I hope!) enjoy using our QLs, QPCs, QXLs, Ataris or whatever. But how many of us bother to pay attention to the health and safety aspects of using our beloved computers? In the workplace, we'd probably all complain like mad if we found a VDU facing a window, or a hopelessly uncomfortable workstation, a dirty screen or similar problems. But in the home environment we tend to forget about such things, or at least not worry so much about them.

In this issue, Geoff Wicks begins a short series on good working practices when using computers and VDUs, gained from his years of experience working in the Netherlands. While rules, regulations and working practices may vary from country to country, the 'proper way' of doing things is often a matter of common sense or just following simple guidelines. And often, small changes to working practices may make quite a difference to your comfort, eye strain etc when working with computers for a long period.

Right, after reading Geoff's article, I'm off for a 10 minute computer break! My idea of taking a break from my QL is to move to sit at the Aurora to work!

Back again, looking forward to the up and coming show season. Keep an eye on the Show Agenda page at the back of QL Today, and go and visit a QL show in your area soon! You get to know all the news, get to meet most of the traders and best of all get to meet

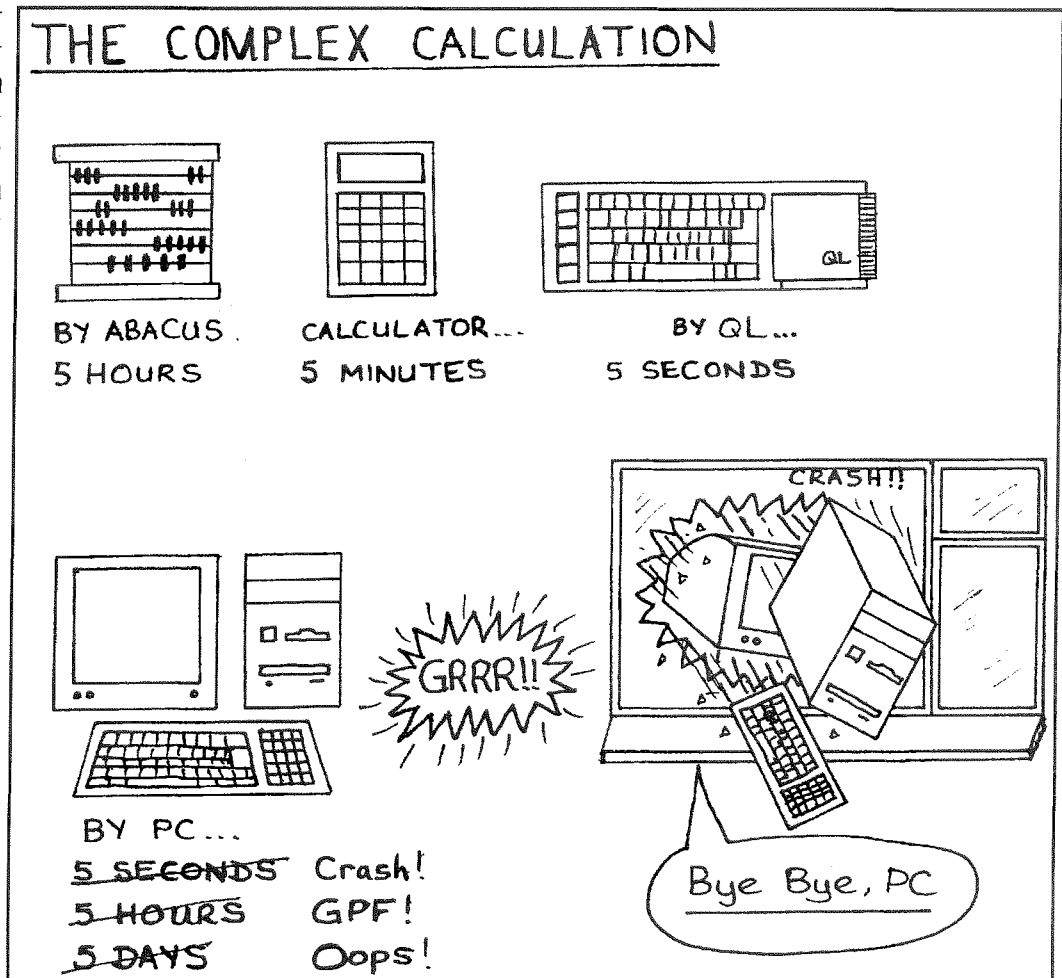
all sorts of QL/Aurora/QPC/QXL users who can help to sort out any problems you may be having with your computer.

Speaking of QL/Aurora/... users, we had quite a few responses to our little competition to find a new collective noun to represent QL users, some light hearted, others quite clever and inventive. You can read some of the suggestions in this issue. A prize will be on its way to someone when a winner has been determined!

More PD software reviews in this issue. Darren Branagh has been doing a wonderful job reviewing PD software (no, he hasn't taken over the magazine, and he does buy commercial software) for us, but is anyone else willing to help him out? We could also do with some reports on some of the software in Quanta's software library - if nothing else it would probably convince QL Today readers who aren't Quanta members that it might be worth joining Quanta just for the large library of software!

All the articles which should have appeared in this issue will come in the next issue - we're at page 60 and every little bit is filled!

Dilwyn Jones



# News

## TF Services - superHermes LITE

superHermes with improved Hermes (serial ports ser1/2 running at full 19200 bps independent of sound, split input baud rates, debounced keyboard and key click) and IBM AT keyboard only.

Currently there is support for following keyboard layouts:

UK, German (2 layouts) , USA (2 layouts), French & Italian (shifted numbers optionally), Danish, Norwegian and Swedish.

Prices are:	superHermes LITE	Kbd lead
UK	£50	£3
EC	£52	£3.50
Europe (outside EC)	£48	£3
Rest of world	£50	£3.50

## PROGS News

We (PROGS) have (finally) moved. The new address is:

Dr. Fr. Hemerijckxlaan 13  
B-2650 Edegem  
tel: 03/457 84 88  
fax: 03/458 62 07

We have fixed some problems in ProWesS and in PFdata. Also, some improvements in LINEdesign. Most significant are that problems in LINEdesign (though caused by syslib) with printing and with memory loss have been fixed. LINEdesign now also accepts a command line parameter again - it will start with the given file loaded.

In ProWesS there have been some small improvements (e.g. the aforementioned syslib bugs). In the last few months there have been somewhat significant improvements in the menu type. Also a new listselect type has been added. There have also been some small improvements in the installation process.

We are currently working on a file management program which will run under ProWesS. However, with the amount of progress that there has been in the last few weeks, we don't expect this to be finished before the end of September.

## QBranch News

**Page Designer 3:** Barry Ansell is working on a 'wide - screen' version of PD 3 for users of the Aurora, Atari and QXL machines with higher resolution. Since Page Designer 3 originally

came out, it has become clear that users would like support for the extended display modes now commonly available on QL-compatibles. Present versions have some support to allow them to work on extended resolutions, but no display resize is possible, for example.

**QXLWIN Reader:** QBranch hope to release fairly soon a program which will enable you to transfer files between the QXL.WIN files and Windows. The author has already submitted a working version to QBranch and it is currently being tested to see if it is ready and suitable for release.

## Ultra Gold Card

Miracle Systems have announced that they are now working on a new card to give the QL another boost in performance. The ULTRA GOLD CARD will be based around a 66MHz 68060 processor and is expected to give an eight fold improvement in speed over the SUPER GOLD CARD. It will require a SUPER GOLD CARD in the system to handle the I/O and will plug into the Q-Plane as a peripheral and will be compatible with Aurora. There will be a SIMM socket to allow for large amounts of RAM and a high speed network. The unit is expected to be available later this year and will be produced by Q-BRANCH.

## News from Jonathan Hudson

**unic 0.06:** Converts PIC files to GIF, PNG, (portable network graphics), TIFF, Postscript, BMP, PCX. Now has graphical front end with 'Save As' option for above formats (except BMP).

**Lynx 2.3.7:** Text based HTML browser, with 'hooks' to display GIF etc. Excellent to convert HTML to ASCII for further viewing or printing. This version of lynx can only view HTML files from local disk, due to the lack of a suitable socket library for QDOS. In the event of such a library becoming available, then a simple relinking would enable lynx to perform as a fully functional internet client, with WWW, ftp, mail, news, wais, gopher etc.

Both available from

<http://www.jrhudson.demon.co.uk>, at least.

## News from Steve Johnson

Steve Johnson has announced further additions to his already extensive collection of QL PD and Freeware software. Three new general collections have been released.

SJPD72 contains QEyes, Snapshot Convert (a program to convert Spectrum emulator snapshot files), Unpic V002 (Jonathan Hudson's program for converting QL PIC files into a variety of PC graphics file formats), ER Dictionary (explains all the medical jargon used in the popular TV series ER), and Scratch, a "drag and drop" active icon for the button frame.

SJPD73 contains a demo version of JMS's QSpread spreadsheet for the QL, Info-Zip Unzip v5.3, and the XPR v3.56 libraries, used by QTPI and QeM.

SJPD74 contains Winds of Change, a pair of programs for easy repositioning of BASIC's channels 0, 1 and 2 windows, and for helping The Editor to use the extended screen resolutions now available. PEX32 is an updated utility allowing i/o action on buried windows (needs, Minerva, Pointer Environment or SMSQ or ATARI emulator). File Selector is a program to aid with getting around a bug in one version of Line-Design which prevented it printing when the button frame was used. VAT Calculator - a powerful and configurable Value Added Tax calculator. FileInfo v3.02 is the latest version of this useful utility for use with QPAC2. Mines is a German version of a similar game on a certain computer (not telling you which, but the clue is that it's not a popular machine with the QL Today team!), Editor v1.82 is a small but powerful text editor (in German). Memv is a small utility to display a memory usage bar on the left side of the QL screen, sadly not Aurora compatible. Finally for this disk, ACP v3.4 is the latest version of Archivers Control Panel.

The specials section now includes SJS182, a set of specialised QTyp dictionaries, covering such subjects as medical, law and Bible terminology. Steve asks that donations be made to The National Back Pain Association for use of these disks (he has suffered greatly from back problems himself, as regular readers will know).

SJS183 contains the latest version of Tim Swenson's Z88 Sourcebook and some Z88 programs. QL Today hopes to provide an updated review of this electronic book soon, as it has been greatly enhanced since our review of an earlier edition was printed.

SJS184 contains Bruce Nicholls' Screen Dazzler screen saver utility, now generously donated to the PD scene as Freeware. If you enjoy using Screen Dazzler, SJS185 also contains an extensive set of additional screen saver routines by Alan Ingrey and Dilwyn Jones (who he?).

SJS186 and 187 respectively contain the original set of clipart and fonts supplied with Line-Design (the current version is supplied without clipart and fonts to reduce costs, you can get the clipart and fonts from PD libraries, for example). While on the subject of PROGS, Steve tells me that the ProWesS documentation disks on SJS160 had a major update on 9/7/97. SJS174 (Gwass etc.) has also had a major update.

The Classic Books section has also had some significant additions since we last brought you news of it.

CB178 contains the Pathology of Lying etc. by William and Mary Heale, CB179 is a 3 disk set containing Lorna Doone, A Romance of Exmoor. CB180 holds The Black Arrow by Robert Louis Stevenson, while CB181 holds another Stevenson work, The Master of Ballantrae. CB182 (2 disks) contains The Complete English Grammar Course. CB183 (2 disks) has the works of W.B. Yeats and CB184 is the works of John Keats. The works of Hippocrates are on CB185. The epic Ben-Hur, A Tale of the Christ, is on CB186. CB187 contains a childhood favourite of mine, The Merry Adventures of Robin Hood by Howard Pyle. Charles Dickens' Little Dorrit (a 4 disk package) can be found on CB188. CB189 is a disk about the History of Britain. Alexandre Dumas' The Black Tulip is on CB190, while CB191 chronicles The Life of Thomas Telford, the civil engineer renowned for his works on the roads and bridges in Britain. You can find over 200 recipes for Italian dishes in The Cook's Decameron on CB192. One thing troubles me about the classic books collection - where on earth does Steve Johnson get the time to read all these?

Steve's email address has changed during the summer. Almac BBS has closed down, so he can no longer be emailed at the old ALMAC address. Customers can now use

**SteveJohnson2@compuserve.com**  
to email him.

### Jochen Merz Software

QMenu 7.03 offers even more useful features for saving files. The file-select menu was designed for choosing a file to load as comfortable as possible. With features introduced in V7, it became useful to select filenames to save something to. Now you can change the destination device with one mouseclick: if the suggested filename is, say, WIN1\_fred\_bas, but you want to save it to

RAM-Disk instead, then all you have to do NOW (in V7.03) is: DO the device name or number (i.e. select them with the ENTER-key or right mouse-button. This will not only change the directory window, but will also apply the selected device name or number to the suggested filename. Quite easy, and very handy! Registered QMENU/QD/QSpread users can download an update free from either BBS. It will also come with every new product containing the Menu extension, of course.

QPC is available in Version 1.30. Once again, Marcel has managed to speed it up considerably (my tests have shown speed increases between 10 and 20%!!!). Again, registered users can get it from the BBSs or via ordinary post.

No news on the colour drivers yet. It seems that November could be a realistic date to see something working. It is much more work than you may think: not only all of the 50 screen driver traps (including panning, scrolling, printing characters etc.) have to be implemented, a way of selecting more colours has to be found. Not to forget, that all parts of the Pointer Environment have to be rewritten as well: sprite handling, pointers, save areas, and so on. QL Today will keep you informed.

## News from the Meeting in Solms

Peter Graf has shown a very interesting looking mainboard:

### Q40 - Preliminary information

The Q40 hardware replaces the Sinclair QL delivering much more speed, graphics, storage and peripheral capabilities. It uses a completely new mainboard, designed to run QDOS on a 68040.

Q40 Mainboard Like the original QL mainboard the Q40 is a complete solution including graphics, peripherals and sound. It fits directly into an industry standard case. No parts from an existing QL, no CPU card, no backplane, no keyboard interface are needed. The Q40 mainboard is equipped with a Motorola M68040 CPU, probably running at 33 or 40MHz. Here is a list of features:

RAM EDO DRAM 4MB up to 32MB. Fast EDO Burst access. Two 72 pin PS2-SIMM sockets for easy upgrading.

ROM 256KB, data bus 32 bit wide. Up to 1 MB are supported.

Graphics High speed graphics chipset and 32 bit dual-port video RAM. Video modes:

- QL mode 8 256x256 pixels, 8 colours

- QL mode 4 512x256 pixels, 4 colours
- Hi-colour mode 512x256 pixels, 64k colours
- Hi-colour mode 1024x512 pixels, 64k colours

Multisync monitor output, 65Hz vertical refresh rate, non-interlaced, dual screen capability.  
Keyboard Interface Controller for MF-102 keyboards.

Sound Stereo digital-analog converters, 10kHz or 20kHz sample rate.

Clock and Nonvolatile RAM Battery buffered real time clock 2KB nonvolatile RAM

Disk Interface Enhanced IDE controller for up to 4 harddisks (16 bit wide data transfer) Floppy disk controller for 2 HD floppies

Ports 2 serial ports with FIFO, 57600 Baud 1 parallel port 1 joystick port Disk interface and ports are on a compact IDE/Multi IO card that belongs to the mainboard

Extension slot The mainboard has two extension slots, one of them is occupied by the IDE/Multi IO card. There are 16 data lines and 20 address lines, all signals are buffered and well defined. Signals and pinout also allow some ISA cards to be used in the slot.

Q40 System A complete Q40 system is a Q40 mainboard in combination with IDE harddisk and HD floppy mounted in an industry standard case with power supply, and an MF-102 keyboard (Microdrives and slow QL network are not supported).

Mainboard prototype is in the state of hardware testing.

For more infos, or if you want to offer help:

Dipl.-Ing. Peter Graf, Lahnstr. 32, 35239 Steffenberg, Germany



## Arborescent Sorting

Stephen Poole

Following your article on sorting, here is a listing for my RADIX sort. I don't mind if QL Today wish to use it for its subscribers, but it must not be circulated or used in commercial programs. The program is written in Super-BASIC for the sake of simplicity; I can also write it in SBASIC if required.

The Radix sort routine is a form of tree-sorting (hence the term Arborescent), using binary tree techniques.

Its advantages are:

- 1 - The Radix sort can be used to sort strings, integers or floating point values.

# TF Services

## superHermes

A major hardware upgrade for the QL

- All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound
- IBM AT keyboard interface (plus foreign drivers)
- HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps) Driver for SERIAL MOUSE supplied. Other uses include RTTY/graphics tablet etc
- THREE spare I/O lines (logic) with GND/+5V
- Capslock/scrolllock LED connector
- Turbo/keylock connectors
- 1.5k user data permanently storeable in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software) **£90** (£92/£87/£90)  
 IBM AT UK layout Keyboard ..... **£22** (£24/£23/£27)  
 Serial mouse ..... **£11** (£13/£12/£14)  
 Capslock/scrolllock LED ..... **£1** (£1.50/£1/£1.50)  
 Keyboard or mouse lead ..... **£3** (£3.50/£3/£3.50)  
 High speed serial (ser3) lead ..... **£4** (£4.50/£4/£4.50)

Hermes available for **£25** (£26/£24/£27) (w/king ser1/2 and independent input, debounced keyboard & keyclick)

## superHermes LITE

All improved Hermes features (see superHermes) plus an IBM AT keyboard interface only. First entry superHermes, & upgradeable to full superHermes.

Cost (incl keyboard lead) .. **£53** (£55.50/£51/£53.50)  
 (specify whether in-line or panel mounted lead)

## I2C INTERFACES

Connects to Minerva and any Philips I<sup>2</sup>C bus

**Power Driver Interface** Similar to parallel below (16 I/O logic lines) except that 12 logic lines can be used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors) ... **£40** (£43/£38/£44)

4 amp total (for motors etc) ..... **£45** (£48/£43/£50)

**Relays** (Eight boxed 3 amp 12v 2-way mains relays connecting to 2 amp power driver ..... **£25** (£28/£23/£27)

**Parallel Interface** Gives 16 input/output lines. Can be used wherever logic signals are required. **£25** (£28/£23/£27)

**Analogue Interface** Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temperature measurements, sound sampling (to 5 KHz), x/y plotting ..... **£30** (£31.50/£29/£30)

**Temp probe** (-40°C to +125°C) ..... **£10** (£10.50/£10/£11)

**Connector for four temp probes** ..... **£10** (£10.50/£10/£11)

**Data sheets** ..... **£2** (£2.50/£2/£3)

## Minerva

The ORIGINAL system operating system upgrade

MINERVA RTC (MKII) + battery for 256 bytes ram.  
 CRASHPROOF clock & I<sup>2</sup>C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast restate. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic. 1.98 now out supporting Qshang.

First upgrade free. Otherwise send £3 (+£5 for manual if reqd). Send disk plus SAE or two IRCs.

MK1...£40 (£41/£40/£43) MKII...£65 (£66/£63/£67)

## QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

**£27** including 6 month guarantee

## QL SPARES

Keyboard membrane ..... **£12** (£12.50/£12/£13.50)

1377 PAL ..... **£3** (£3.50/£3/£4)

Circuit diagrams ..... **£3** (£3.50/£3/£4)

68008 cpu or 8049 IPC ..... **£8** (£8.50/£7.50/£9)

8301/8302 or JM ROM set ..... **£10** (£10.50/£10/£11)

Serial lead ..... **£10** (£11.50/£11/£12)

Power supply (sea mail overseas) .... **£12** (£17/£16/£21)

Other components (sockets etc) also available

## QBBS Bulletin Board

UKs first QL scrolling bulletin board

Megabytes of files. Messages to/from UK/Belgium/Holland/USA/Italy/Germany

**01344-890987 (up to V34 28800 bps)**

## COMPUTER CLEANERS (UK only)

All in standard 3-pin devices and are plug-in NO WIRING REQUIRED. In their twelfth year of production - as old as the Sinclair QL

2-way adaptor..£14      3-way adaptor.. £18

4-way trailing socket.... £24

AUG 97

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). Send SAE or IRC for full list and details

VISA

Holly Corner, Priory Road, Chavey Down, ASCOT, Berks, SL5 8RL  
 Tel: 01344-890986 Fax/BBS: 01344-890987  
 tony@firshman.demon.co.uk http://www.firshman.demon.co.uk

MasterCard

2 - It is faster than Quicksort 1 in the sorting routines article, but marginally slower than Quicksort 2.

3 - It does not use up stack space and can, therefore, sort very large arrays, of up to 32766 items on a Super Gold Card.

Line by line description:

100: The routine is called a Radix sort since the root of the tree is a median value.

110: 'ch' is the number of characters per word. 't' is the maximum number of words. 'pr' is the switch to turn printing on or off

210 to 240: sets up a randomly filled array for the purposes of demonstration

260 to 340: this fills up a binary tree. The routine compares each string to nodes of the tree. If the string is smaller, it takes a left fork, else a right fork. When it finds a free space, then it makes a note in the node where the next smaller and larger strings are to be found, filling the index

360 to 430: When the tree is full, the pointer array is used. The whole tree is traversed from the smallest string up to the largest one, and the then sorted string is stored in 's\$'

450 to 470: This calculates the time to fill the tree, and then to sort using the index.

```
100 REMark RADIX_sort by S.Poole, v25jun97
110 ch=4: t=32766: pr=1
120 DIM s$(t,ch),t$(t,ch),t%(t,3),w$(ch+1),x$(ch+1)
130 WINDOW 480,88,16,16: BORDER 0: CSIZE 1,0: PAPER 0: INK 7: OVER 1: CLS
140 :
150 a=-1: b=1: d=2: u=3: aa=CODE('a'): zz=CODE('z')
160 REPEAT loop
170 INPUT \How many words? :!w$: w$='0'&w$
180 L$=t-1: IF w$<'2' OR w$>L$: NEXT loop: END IF
190 w=w$: w1=w+1: DIM t%(0,0),t%(t,3): t$(1)='mnmnmnmn': IF pr: PRINT
200 :
210 FOR f=2 TO w1
220 w$='': FOR j=1 TO ch: r=RND(aa TO zz): w$=w$&CHR$(r): END FOR j
230 t$(f)=w$: IF pr: PRINT w$&' ':
240 END FOR f: d1=DATE
250 :
260 FOR f=2 TO w1
270 w$t$(f): p=1
280 REPEAT q
290 x$t$(p): z=(w$>x$): z=2+z: r=t%(p,z)
300 IF r: p=r: NEXT q: END IF
310 EXIT q
320 END REPEAT q
330 t%(p,z)=f: t%(f,b)=p
340 END FOR f
350 :
360 n=0: p=1: ct=0: d2=DATE: IF pr: PRINT
370 REPEAT q
380 l=t%(p,d): IF (l>0): t%(p,d)=a: p=1: NEXT q: END IF
390 r=t%(p,u): IF (r= a): k=t%(p,b): t%(p,b)=a: p=k: NEXT q: END IF
400 IF (p<>'1'): ct=ct+1: s$(ct)=t$(p): IF pr: PRINT t$(p)&' ': END IF : END IF
410 n=n+1: IF n=w1: EXIT q: END IF
420 IF r: t%(p,u)=a: p=r: NEXT q: ELSE t%(p,r)=a: p=t%(p,b): END IF
430 END REPEAT q
440 :
450 d3=d2-d1: d4=DATE: d5=d4-d2: d6=d3+d5
460 IF d6: d7=w/d6: ELSE d7=w: END IF
470 PRINT: PRINT !INT(d3)!INT(d5)!!INT(d7)!' words/sec'
480 END REPEAT loop
```



# Display Code

Dilwyn Jones

This is a small suite of 9 basic extensions written in assembler designed to help SuperBASIC or SBASIC programmers cope with extended display modes on more recent hardware and emulators such as Aurora, QPC and QXL.

Although extensions are built into SMSQ allowing easy checking of such details as screen size and location in memory, programs using those extensions are limited to running on systems with SMSQ and SBASIC.

These extensions will work on SMSQ and QDOS, providing a means of consistently returning the required information, allowing programs a means of working on all systems. Graphical applications often need to write direct to memory, or at least to know the screen size and location details. That's the sort of information this code will allow you to extract from the system.

Over the years, many programs were written which were later found not to work on displays other than the original 512x256 QL screen. The forward-thinking designers of the QL had actually allowed for the possibility of larger screen sizes by including information in the system which was available to machine code programs, but not to SuperBASIC programmers. As the information about this was not readily documented and available or widely understood in those early days, many programs just assumed the original QL display and hence could not work properly when the size and location of the screen memory changed - I know, I wrote such programs myself! This set of extensions won't fix those early programs of course, but does give a simple means of extracting this information for SuperBASIC and SBASIC programmers so that new programs at least won't be guilty of the same sins. To be fair, with the benefit of hindsight it is easy to refer to those old programs as being sinful, though at the time the necessary information was neither readily accessible nor widely understood.

The file was on issue 2's cover disk - look for DISPLAY\_CDE. You can simply

```
LRESPR FLP1_DISPLAY_CDE
```

if Toolkit 2's LRESPR command is available (remember it may give a 'not complete' error if there are any jobs running at the time), or add

this to your boot program:

```
base=RESPR(598):  
LBYTES FLP1_DISPLAY_CDE,base:  
CALL base
```

I hope that the names I have given the extensions don't clash with anything you already have installed on your system. If there is a clash, you'll need to either hack the names of the extensions in DISPLAY\_CDE using your preferred editor, or reassemble the source code file DISPLAY\_ASM after altering the names in the dc.b statements in the table.

There are eight functions and one procedure:

## ADDRESS

```
LET adr = ADDRESS(#channel)
```

This function returns the base address of the display for the given window channel. Normally you'd use #0. For a 512x256. Normally, you'd only need this value if you were writing programs which wrote directly to the screen area, e.g. using LBYTES to load a screen direct to the video area of memory, using a command such as

```
LBYTES filename,ADDRESS(#0)
```

assuming the screen being loaded was the same size as the current display.

## BYTES

```
LET bytes_per_line = BYTES(#channel)
```

The display is organised as a series of horizontal lines, with each line being a given number of bytes wide. In several display sizes, the exact width of these lines in bytes happens to be the number of pixels DIV 4, but this is a dangerous assumption to make - many programs made this assumption and fell over when the Aurora came along, as some of its display modes use a fixed line width, irrespective of the number of pixels on a line, meaning that some of the bytes used to store each line are actually unused. So if you are writing individual lines to the screen, as you would for video effects for example, you need to take account of how many bytes there are between the start of one line and the next. That's the purpose of this function - it tells you how many bytes lie between where one line starts and the next line starts. Users of version JM or earlier ROMs should note that this information is not available, as the area which holds this value is used for something else, so a version JM machine always assumes it has a 128 byte line width.

## DMODE

```
LET display_mode = DMODE
```

Returns the mode number of the current display. This would usually be 0 for the 4 colour modes, and 8 for the 8 colour modes. As the routine uses the mt.inf system trap, it ought to handle the extended colour mode drivers, or monochrome modes on certain emulators (both cases untested at the time of writing). I am not sure what will happen if this function is used while one of the old mixed mode screen displays are used (there are extensions in Quanta library I think which allow part of the screen to be in MODE 4 and part in MODE 8, for example)

## SYS\_VAR

```
LET system_vars = SYS_VAR
```

Tells you at what address you can find the system variables. Now you can PEEK and POKE to your heart's content if you really need to!!!

The FLIM\_n extensions return information about the maximum sizes or limits of a screen window size. As it uses the iop.flim trap, it means the information can't be extracted if this is not implemented on your system. But I think I'm right in assuming that if iop.flim is not on the system for whatever reason, the system can't use extended displays anyhow. If it can't get the required information, it assumes you're running a 512x256 QL screen rather than unhelpfully causing an error report. If you supply a primary channel window, the values returned will be the maximum possible size for that window (essentially the full display width and height), whereas if you supply a secondary channel number the values returned refer to the outline area for the primary. If you don't know what this means, supply the lowest window channel number opened, e.g. #0 for BASIC. The first two extensions return origin details, whereas the other two return the width and height details.

### FLIM\_X

```
LET x_origin = FLIM_X(#channel)
```

### FLIM\_Y

```
LET y_origin = FLIM_Y(#channel)
```

### FLIM\_W

```
LET wide = FLIM_W(#channel)
```

### FLIM\_H

```
LET high = FLIM_H(#channel)
```

```
MOVEMEM from_address, to_address,  
number_of_bytes
```

This procedure lets you move the content of memory around. Simply tell it where to move from, where to move it to, and how many bytes. Negative values will cause errors. There is no check on overlaps etc. so with care you can use this to fill memory areas by writing the first byte value with a POKE, for example, then moving this up to fill the required number of bytes. It always moves from lower addresses first - there is nothing particularly intelligent about this command in terms of working out the best way to move things. It is quite slow by comparison with similar commands in other toolkits.

## Examples of Use

It may be more instructive to list a few simple examples of how to use these extensions for simple applications. Note that they use the Toolkit 2 extensions ALCHP and RECHP for allocating and deallocating temporary buffer areas in the common heap area of memory. Most systems these days have Toolkit 2 or equivalent commands so this should not be a problem.

### 1. FULL SCREEN WINDOW

This routine shows you how to set a window to occupy the full screen, or the full outline area available to it if it is a secondary window. To make channel #0 occupy the full area of the screen, enter the command FULL\_SCREEN #0.

If you tried the same thing on channel #2 on a VGA display, for example, #2 would be set to the maximum possible area as covered by the outline for the primary channel, in this case, #0, which would normally cover #0 and #1 and #2 in BASIC.

The procedure leaves the actual values in the variables dw and dh (width and height) and dx and dy (origin co-ordinates). Note that this routine doesn't actually do anything visibly, you may need to do a CLS command on the channel concerned, for example, to see its effect.

### 2. STORE A SCREEN IN MEMORY

This routine sets up an area in the common heap to store a copy of the current screen. The address of this area is given by the variable "area". We work out the start address of where to copy from the screen using the ADDRESS

function, and the total number of bytes to copy is calculated by the product of the number of bytes per line (given by the BYTES function) and the height of the screen (given by the FLIM\_H function). Note that when using 512x256 mode on the Aurora, for example, this routine will save the whole memory used for the screen, not just the visible area, as Aurora uses a fixed line length unrelated to the actual number of pixels used on the current display, meaning that although you only see 512 pixels across, for example, the line used to hold this display is 1024 bytes wide, but only 512 used and visible, so the calculation is not as obvious as might be thought at first. A typical application of this little routine might be to store a graphical screen in memory while a menu is superimposed on the picture. The variable "screen\_length" holds the actual length (in bytes) of the screen saved.

### 3. RESTORE A SCREEN FROM MEMORY

This routine restores the screen saved by the previous procedure, and releases the memory area used to store it, by using the RECHP command from Toolkit 2.

### 4. MERGE SCREEN

There is a large number of clipart screens available for the QL, mostly as 512x256 QL screens. In the old days, when each QL had the same size screen, it was easy enough to load these direct to the screen with a simple

```
LBYTES filename,131072
```

command. Not only doesn't this work on modern large displays, it might actually crash the system in some cases, since the area of memory previously used by the screen may now be used by something else. This routine tackles this problem by loading the 32k (512x256 pixels) screens into a buffer area in the common heap, then copies it line by line into the top left corner of the display. Note how two variables are used to keep track of where each line starts. With old 512x256 screens, we know they are 128 bytes wide, so it is easy enough to step through them 128 bytes at a time. The other variable is incremented by the width of each display line, given by the BYTES function. Of course, writing direct to the screen is not the done thing, and the picture may well be ruined if another job is writing to the screen at the same time! Finally, when the transfer is complete, the heap memory is released with the RECHP command.

### 4. FILL MEMORY

A task which arises now and again in programming is to fill a given area of memory with a particular value. This routine takes advantage of how the MOVEMEM command works. The command should be issued in this form:  
`FILL_MEM start_address, how_many_bytes, what_value`

The routine works by setting the first byte of the area to be filled, using the POKE command. Then, it copies this up one byte with the MOVEMEM command, which repeatedly copies each byte up one address, thus the byte copied is always the value of the previous byte and the area is filled with the value of the first byte fairly quickly.

Another example: if you wished to turn the entire display black, then you could issue the following command. This is quite a naughty way of doing things, but it serves to illustrate how the command works:

```
FILL_MEM ADDRESS(#0), BYTES(#0)*FLIM_H(#0),0
```

### 5. SYSTEM\_VALUE

This routine reads a value from the system variables. You don't need to supply the absolute address, just the offset as published in several QL technical manuals. The routine adds the offset to the base address, peeks a value from there and returns it as the value of the function.

```
LET value = SYSTEM_VALUE(offset)
```

For example,

```
PRINT SYSTEM_VALUE (140)
```

prints the value of the auto repeat delay, while

```
PRINT SYSTEM_VALUE(55)
```

prints the network station number.

### 6. SET MODE

Some programs which need to switch between 4 colour and 8 colour mode often set the screen mode (causing an irritating flashing) even if the screen was already in the required mode. A short routine like this can check the current mode and only change it if it is the wrong mode, thus preventing the flashing of windows you get when changing mode to the same mode.

The following pages contain the BASIC listing, the Assembler source - and don't forget, you already had the ready-to-LRESPP extension on the cover disk of the previous issue!

## The BASIC-Listing

```
100 REMark example routines for use with DISPLAY_CDE
110 REMark written by Dilwyn Jones, June 1997
120 :
1000 DEFine PROCedure FULL_SCREEN (channel)
1010   dw = FLIM_W(#channel) : dh = FLIM_H(#channel)
1020   dx = FLIM_X(#channel) : dy = FLIM_Y(#channel)
1030   WINDOW #channel,dw,dh,dx,dy
1040 END DEFine FULL_SCREEN
1050 :
1060 DEFine PROCedure STORE_A_SCREEN
1070   screen_length = BYTES(#0)*FLIM_H(#0)
1080   area = ALCHP(screen_length)
1090   IF area < 0 THEN PRINT #0,'Unable to reserve memory.' : RETURN
1100   MOVEMEM ADDRESS(#0) TO area, screen_length
1110 END DEFine STORE_A_SCREEN
1120 :
1130 DEFine PROCedure RESTORE_A_SCREEN
1140   IF area > 0 THEN
1150     MOVEMEM area TO ADDRESS(#0), screen_length
1160   END IF
1170   RECHP area
1180 END DEFine RESTORE_A_SCREEN
1190 :
1200 DEFine PROCedure MERGE_SCREEN (filename$)
1210   REMark merges a 32K 512x256 screen onto the top left corner of a larger
display
1220   area = ALCHP (32768)
1230   IF area < 0 THEN PRINT #0,'Unable to allocate memory.' : RETURN
1240   LBYTES filename$,area
1250   from_address = area
1260   to_address = ADDRESS(#0)
1270   FOR a = 1 TO 256
1280     MOVEMEM from_address TO to_address,128
1290     from_address = from_address + 128
1300     to_address = to_address + BYTES(#0)
1310   END FOR a
1320   RECHP area : REMark finished with it
1330 END DEFine MERGE_SCREEN
1340 :
1350 DEFine PROCedure FILL_MEM (addr,no_of_bytes,byte_value)
1360   POKE addr,byte_value
1370   MOVEMEM addr TO addr+1,no_of_bytes-1
1380 END DEFine FILL_MEM
1390 :
1400 DEFine FuNction SYSTEM_VALUE (what_offset)
1410   RETURN PEEK(SYS_VAR+what_offset)
1420 END DEFine SYSTEM_VALUE
1430 :
1440 DEFine PROCedure SET_MODE (mode_number)
1450   IF DMODE <> mode_number THEN MODE mode_number
1460 END DEFine SET_MODE
```

We try to keep the size of listings in the magazine as small as possible. As the display handling has become one of the most important issues and we already had a number of enquiries from machine code programmers, we give the assembler source here as well - in smaller print. We hope this is an acceptable compromise between non-machine-code programmers (small size) and machine-code programmers (it is included, even if it is small).

```

* Display Extensions V1.00
* QLiberator directive $$asmb=flp1_display_cde,0,10

* link into basic as extensions
    lea     exts,a1           ;point to list of extensions
    move.w $110,a2          ;bp.init
    jmp     (a2)             ;link extensions

* list of extension names and definitions
exts    dc.w    1           ;just one procedure
        dc.w    movemem-*   ;location
        dc.b    7,'MOVEMEM' ;name
        dc.w    0           ;end of procedures list

        dc.w    8           ;8 functions
        dc.w    address-*   ;location of ADDRESS
        dc.b    7,'ADDRESS' ;name of function
        dc.w    bytes-*     ;location of BYTES
        dc.b    5,'BYTES'   ;name of function
        dc.w    dmode-*     ;location of DMODE
        dc.b    5,'DMODE'   ;name of function
        dc.w    sys_var-*   ;location of SYS_VAR
        dc.b    7,'SYS_VAR' ;name of function
        dc.w    flim_x-*    ;location of FLIM_X
        dc.b    6,'FLIM_X',0 ;name of function
        dc.w    flim_y-*    ;location of FLIM_Y
        dc.b    6,'FLIM_Y',0 ;name of function
        dc.w    flim_w-*    ;location of FLIM_W
        dc.b    6,'FLIM_W',0 ;name of function
        dc.w    flim_h-*    ;location of FLIM_H
        dc.b    6,'FLIM_H',0 ;name of function
        dc.w    0           ;end of function definitions list

* code for the MOVEMEM procedure
* MOVEMEM from_address,to_address,bytes
movemem move.w $118,a2      ;ca.gtlin to fetch long integers
        jsr     (a2)        ;try to fetch 3 long integers
        bne    return      ;oops
        subq.w #3,d3        ;3 parameters?
        bne    badparam    ;oops

*check parameter values that addresses and bytes are positive
        tst.l  0(a6,a1.l)   ;source address
        blt    badparam    ;invalid negative address?
        move.l 0(a6,a1.l),a2 ;source address into a2
        tst.l  4(a6,a1.l)   ;destination address
        blt    badparam    ;invalid negative address
        move.l 4(a6,a1.l),a3 ;destination address into a3
        tst.l  8(a6,a1.l)   ;number of bytes
        beq.s  mvedone      ;zero bytes, don't do anything
        blt    badparam    ;invalid negative number of bytes
        move.l 8(a6,a1.l),d1 ;number of bytes to be moved
movelop move.b (a2)+,(a3)+ ;shift a byte
        subq.l #1,d1        ;count the bytes
        bne.s  movelop     ;keep going until all done
mvedone moveq #0,d0        ;no error report
        rts               ;return to basic

* LET adr = ADDRESS(#channel)
* returns base address for given screen channel
address move.w $118,a2      ;ca.gtlin
        jsr     (a2)        ;fetch 1 parameter
        bne    return      ;oops...
        subq.w #1,d3        ;1 parameter?
        bne    badparam    ;oops...

* fine, we have one parameter on the stack, so for any
* further errors, we can return a suitable negative number
* instead of an error report
* all the functions use d7 to indicate how many bytes more are
* required on top of any parameters already on the RI stack (the
* channel number is a 4 byte value, so the norm is 2 extra bytes
* needed to return a 6 byte float value)
        moveq  #2,d7        ;2 extra bytes will be needed

```

```

    move.l 0(a6,a1.l),d0 ;get channel number to d0
    bmi   retfp1         ;return the negative d0 value
    bsr   chan2id        ;convert d0 channel# to ID in a0
    bmi   retfp1         ;oops, return the negative number

    moveq  #9,d0         ;prepare to use the channel

    moveq  #$32,d1       ;sd.scrb offset within definition block
    moveq  #-1,d3        ;infinite timeout
    lea.l  extop_1,a2    ;the routine to be invoked by sd.extop
    trap  #3             ;use sd.extop to call the routine
* d1 now contains the value required (screen base address)
    move.l d1,d0         ;put required value in d0 for retfp
    bra   retfp1         ;and return it

* this routine is invoked from sd.extop to fetch a value indexed by d1
* from the base of the definition block in a0
* (Thank you Norman Dunbar for originally showing me how to use
* sd.extop)
extop_l   move.l 0(a0,d1.w),d1 ;fetch long value required
          moveq  #0,d0         ;ensure no error
          rts                 ;and go back to caller
extop_w   move.w 0(a0,d1.w),d1 ;fetch word value required
          moveq  #0,d0         ;ensure no error
          rts                 ;and go back to caller

* LET bytes_per_line = BYTES(#channel)
* returns display width for given screen channel
* AH/JM roms (QDOS 1.03 or earlier) will return wrong result
* so return fixed value of 128 for these roms
bytes     move.w $118,a2      ;ca.gtlin
          jsr   (a2)          ;fetch 1 parameter
          bne  return        ;oops
          subq.w #1,d3        ;1 parameter?
          bne  badparam      ;oops

* fine, we have one parameter on the stack, so for any further errors
* we can return a suitable negative number instead of an error report

* all the functions use d7 to indicate how many bytes more are required
* on top of any parameters already on the RI stack (the channel number
* is a 4 byte value, so the norm is 2 extra bytes needed to return a
* 6 byte float value)
          moveq  #2,d7        ;2 extra bytes will be needed
          move.l 0(a6,a1.l),d0 ;get channel number into d0
          bmi   retfp1         ;return the negative d0 value
          bsr   chan2id        ;convert channel# in d0 to ID in a0
          bmi   retfp1         ;oops, return the negative number

* at this point:
* A0=channel ID  A1=RI ptr  d7=extra RI stack bytes wanted

* Thanks to Ralf Rekoendt for this tip originally
* test if AH or JM rom (QDOS 1.03 or lower) and if so, return 128
* the only value supported by these rom versions (fixed line width of
* 128 bytes wide only)
* need to preserve channel ID in a0, which is smashed by mt.inf
          move.l a0,d6        ;save channel id which was in a0
          moveq  #0,d0        ;call mt.inf to get system information
          trap  #1            ;call mt.inf
          move.l d6,a0        ;return channel ID to a0
* d2.l now contains the ASCII version number (n.nn)
* remove the decimal point to make things easier
* as some national ROM versions contain a letter code rather than "."
          move.l #128,d0      ;in case it's an AH/JM rom...
          and.l  #$FF00FFFF,d2 ;remove '.'
          cmpi.l #$31003034,d2 ;>=1.04? <"1">,<nul>,<"0">,<"4">
          bcs   retfp1        ;no, return 128

* prepare to fetch info from definition block
          moveq  #9,d0         ;prepare to use the channel
          moveq  #$64,d1       ;offset in chan def block for sd.linel
          moveq  #-1,d3        ;infinite timeout
          lea.l  extop_w,a2    ;the routine to be called by sd.extop

```

```

        trap    #3                ;invoke sd.extop
* d1 now contains the value required (screen base)
        ext.l   d1                ;stretch to long word
        move.l  d1,d0            ;put required value in d0 for retfp
        bra    retfp1           ;and return it

* LET mde = DMODE
* no parameters in the basic call; return 0 or 8 for mode number
dmode  move.l  a5,d0            ;pointer to last parameter
        sub.l   a3,d0            ;subtract pointer to parameter 1
        lsr.l  #3,d0            ;bytes DIV 8 = number of parameters
        bne    badparam         ;oops, silly boy
        moveq  #$10,d0          ;mt.dmode
        move.b #-1,d1            ;READ mode
        move.b #-1,d2            ;READ display (incidental here)
        trap   #1
* d1.b=display mode number
        moveq.l #6,d7            ;6 bytes needed on stack to retn float
        clr.l   d0                ;empty top 3 bytes
        move.b  d1,d0            ;mode number for conversion to fp
        bra    retfp            ;send it back to basic

* LET sys_variables = SYS_VAR
sys_var move.l  a5,d0            ;pointer to last parameter
        sub.l   a3,d0            ;subtract pointer to 1st parameter
        lsr.l  #3,d0            ;number of parameters (8 bytes/param)0
        bne    badparam         ;oops
        moveq.l #0,d0            ;mt.inf
        trap   #1                ;call mt.inf
* returns d2.l = QDOS ASCII version number n.nn
* and a0 = pointer to system variables
        move.l  a0,d0            ;system variables to d0
        moveq.l #6,d7            ;space needed
        bra.s  retfp            ;return it to basic

* the iop.flim based calls follow
* LET w=FLIM_W(#channel)        LET h=FLIM_H(#channel)
* LET x=FLIM_X(#channel)        LET y=FLIM_Y(#channel)
* routines need a 4 byte storage area required by iop.flim for
* width,height,x,y
store  ds.w    4
flim_w move.w  #512,d5           ;default width
        moveq.l #0,d6            ;offset of width value from "store"
        bra.s  flim             ;branch to common part of routine

flim_h move.w  #256,d5           ;default height
        moveq.l #2,d6            ;offset of height value from "store"
        bra.s  flim             ;branch to common part of routine

flim_x move.w  #0,d5             ;default x
        moveq.l #4,d6            ;offset of x value from "store"
        bra.s  flim             ;branch to common part of routine

flim_y move.w  #0,d5             ;default y
        moveq.l #6,d6            ;offset of y value from "store"

* common routine, with pre-assigned values in d5+d6
flim   move.w  $118,a2           ;ca.gtlin
        jsr    (a2)              ;fetch 1 parameter
        bne.s  return           ;oops!
        subq.w #1,d3             ;1 parameter
        bne.s  badparam         ;oops!
* we now have the channel number on the stack
        moveq.l #2,d7            ;2 more bytes will be needed
        move.l  0(a6,a1.l),d0     ;get channel number to d0
        bmi.s  retfp1           ;if negative, return as function value
        bsr    chan2id          ;channel # to ID in a0
        bmi.s  retfp1           ;oops,return the -ve number as fn value

        moveq  #-1,d3            ;timeout
        clr.l   d2                ;has to be zero
        move.l  a1,$58(a6)       ;store RI stack pointer

```

```

        lea.l   store,a1           ;point to 8 byte area
        moveq  #$6C,d0           ;iop.flim
        trap   #3
        move.l  $58(a6),a1       ;recover RI stack pointer
        tst.w  d0                ;was there any error?
        beq.s  noerr            ;no, fetch value from store to return
* since iop.flim appears not to exist, we'll assume a 512x256 display
        move.w  d5,d0           ;the default value for this function
        bra.s  extnd           ;and return that default value
* fetch values for return
noerr   lea.l   store,a2         ;point to four word store area
        move.w  0(a2,d6.l),d0    ;fetch value
extnd   ext.l  d0                ;convert to long word
        bra.s  retfp1          ;return as floating point value
* utility routines shared by functions
badparam moveq #-15,d0          ;err.bp
return  rts                    ;go back to basic with d0 error report
notopen moveq #-6,d0           ;err.no
        rts

* convert and return long word in d0 into a floating point number and
* return (used with permission of Simon N. Goodwin).
* d7.w contains no. of extra bytes required on RI stack over and above
* what's on the RI stack parameters
retfp1  move.l  a1,$58(a6)       ;store RI stack pointer if relevant
retfp   move.w  d0,d1            ;d1 will be exponent
        move.l  d0,d2            ;d2 will be mantissa
        beq.s  normalised       ;zero is a trivial case
        move.w  #2079,d1         ;first guess at exponent
        add.l  d0,d0             ;already normalised?
        bvs.s  normalised
        subq.w #1,d1             ;no, halve exponent weight
        move.l  d0,d2            ;double mantissa to match
        moveq  #16,d3            ;try a 16 bit shift
normalise move.l d2,d0           ;take copy of mantissa
        asl.l  d3,d0             ;shift mantissa d3 places
        bvs.s  too_far          ;overflow - must shift less
        sub.w  d3,d1             ;correct exponent for shift
        move.l  d0,d2            ;new mantissa is more normal
too_far asr.w  #1,d3             ;halve shift distance
        bne.s  normalise        ;try shift of 8,4,2 and 1
* is extra space required on RI stack? d7 holds number of bytes
* needed over and above existing parameter space
* need to store anything important not preserved by bv.chrix call
normalised movem.l d1-d2,-(a7)  ;store exponent/mantissa
        tst.l  d7                ;if non-zero, create more space
        beq.s  enough           ;no more needed
        move.l  d7,d1            ;no. of extra bytes needed
        move.w  $11A,a0          ;bv.chrix vector
        jsr    (a0)              ;do it
        move.l  $58(a6),a1       ;recover ri pointer
        sub.w  d7,a1             ;adjust by no. of bytes
        move.l  a1,$58(a6)       ;additional bytes pointed to now
enough  movem.l (a7)+,d1-d2     ;recover exponent/mantissa
        move.l  d2,2(a1,a6.l)    ;stack mantissa
        move.w  d1,0(a1,a6.l)    ;stack exponent
        moveq  #2,d4             ;signal FP result
        moveq  #0,d0             ;no errors PLEASE!
        rts

* subroutine to convert channel # in d0 to 4-byte channel ID in a0
chan2id tst.l  d0                ;can't be negative
        bmi.s  retfp1           ;oops, was negative...
        mulu  #$28,d0           ;convert into channel table offset
        add.l  $30(a6),d0       ;add start of table
        cmp.l  $34(a6),d0       ;past end of table?
        bge.s  notopen         ;yes, channel not open
        movea.l 0(a6,d0.l),a0    ;no-get ID into a0.l
        tst.l  0(a6,d0.l)       ;is channel open?
        bmi.s  notopen         ;no, return err.no
        moveq  #0,d0            ;yes, return OK code
        rts                    ;and exit

```



# QUANTA



## Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:

Monthly Newsletter - up to 40 pages

Massive Software Library - All Free !

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

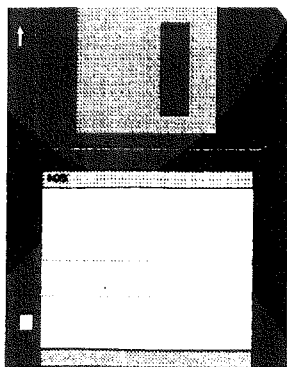
Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

**\*Now in our FOURTEENTH successful year\***

Further details from the Membership Secretary



**Bill Newell**  
**213 Manor Road**  
**Benfleet**  
**Essex**  
**SS7 4JD**  
**Tel. (01268) 754407**



# QPC - My Experiences

Norman Dunbar

I am sure that you have all read about QPC in this magazine and in Quanta. This is another article about it, but don't worry, this is not one of those 'I have QPC and it runs at xyz miles per hour and it does everything I want it to and it is great and why don't you buy it?' articles. This one is one man's experience of installing and using QPC over a number of months. It does not have any benchmarks in it to say how fast it works or anything like that so if that's what you are hoping for, you are about to be slightly disappointed.

Way back in September 1996, I received a package in the post from Dilwyn Jones/Jochen Merz. The envelope contained a manual and a disc for QPC - the QDOS emulator for a PC. As a QXL user I was very interested to see exactly how this software performed and how well it matched up to a 'real' emulator using a 68040 chip.

I must admit that I have not been using QPC constantly since then as it turned out, QPC had a few problems with Windoze 95 and would not install or run. I informed Dilwyn of the problem and he advised me that some other users were having similar problems while others were having no problems at all - just my luck.

A few months went by and another disc arrived with a single sheet of A4 paper. This was an upgrade to QPC, version 2.81 of SMSQ/E and a new set of instructions on installing into Windoze 95. The instructions had been dramatically reduced and even better, simplified to such an extent that even a complete novice could have installed the software - assuming that Windoze 95 hadn't made them give up first!

A very strange thing was noticed about the QPC disc - it is protected. This came as a bit of a surprise as QL software has been unprotected for many years now, although this was not always the case - anyone remember Lenslock or having the original microdrive in MDV1\_? (In fact, does anyone remember microdrives for that matter?) Mind you, this is PC software, not QL!

The protection is such that you must install from the original disc, but you can only install twice. This means that you can have it on your home PC and again at work - if your boss allows it of course (mine does not). If you want to remove QPC from your PC and install it on

another one, you must use the master disc again and uninstall it first. There is obviously a counter somewhere on the master disc which records how often it has been installed and uninstalled so that you can only ever have 2 copies maximum. I personally am not sure about this, but it is up to the author of the software.

This can cause problems, as the manual points out very clearly. For example, never attempt to remove QPC without using the deinstaller or you will lose one of your two 'installations'. Never change the QPC directory on your hard disc without first deinstalling from the current one, and reinstalling to the new one. Never update the SMSQE.BIN file as this must be installed from the master disk, so if a new release of SMSQE.BIN is obtained you have to deinstall, copy the update onto the floppy then reinstall.

The software can of course be run from the master disc without having to install it on your PC's hard disc. I have not tried this method of operation.

QPC comes on a PC format disc, not a QL format. This is obvious when you think about it as the PC would not be able to understand the QL disc format. QPC is, after all, a PC program. There is a copy of Tony Tebby's Config utility on the floppy *it is not the same; written by Marcel, it just DOES the same - Editor!*, this has been converted to PC format so that it runs directly on the PC and can therefore be used to configure the QPC files without having to copy them onto a QL disc first, configure them and finally copy them back to the original disc.

QPC allows you to configure the SMSQE.BIN file using the supplied Config program. The following parts are configurable.

- **Display size** - you are offered a choice of QL 512 by 256, EGA 640 by 350, VGA 640 by 480, SVGA 800 by 600. Or VESA - whatever that is.

- **VESA X, VESA Y, MAX X and MAX Y** options allow you to define a 'custom' screen size. I don't think I have a VESA graphics card and I don't care ! I use EGA as this gives the best results on my monitor.

- **Memory size** - you specify how much of the PC's memory QPC will use. As QPC takes over the whole PC, you are as well to give it the maximum available. If you give it a number higher than the available memory it will use what it finds. The values range from 1 to 16 and are in megabytes. The default seems to be 4 meg.

- **Boot device** - this option lets you specify which device QPC will boot from once running

as a QL, not when run from the PC. You may specify 'none' which will cause it to boot from some other device that you specify next, or 'flp1' or 'flp2' as required.

- **Boot device if 'none'** - allows you to select which hard drive from 1 to 8 QPC will use as its default device. Once again, this is as a QL.

- **PAR printer port** - allows you to choose 'none' or 'LPT1' as your parallel printer port.

- **SER1 serial port** - allows you to select which of the PC's com ports is to be used as ser1. This is to avoid conflicts with PC mouse devices. You also select the IRQ to be used with the port.

- **SER2 serial port** - this is as per the above for SER1.

- **Country code** - SMSQ/E allows the use of country specific error messages etc. This option lets you choose your country.

- **SVGA mode number** - SVGA allows 800 by 600 graphic screens. The mode number was never standardised so some graphics cards on the PC may use a different number to the 'almost standard' 106. You will need to check with your graphics card manual to get the correct setting.

- **Filename for WINx** - Allows you to specify a PC filename for each of the QL hard drives that you want to use. The QXL uses a single file called QXL.WIN on each of the PC's hard drives so win1\_ is C:\QXL.WIN, win2\_ is D:\QXL.WIN and so on. QPC allows you to have up to 8 win devices and they can be positioned anywhere on your PC. You could in fact have them all on the same hard disc. They default to QXL\_WIN on drives C: through J:.

There is also an option to configure the CD audio driver - I haven't changed this.

Having configured your own options, you are ready to roll. But, what if you are not sure about which options to use? Have no fear QPC allows you to use the command line to select your devices etc and thus you can experiment until you figure out the options you need prior to going through the 'test, configure, retest, configure .....' loop. Any values that you don't specify on the command line will be taken from the Config block as normal. This is, I think, an excellent method of starting a QL program and should really be adopted as a 'standard' in the future. *[This is what we do already; QD, QSpread etc. can be configured AND you can pass parameters via command string. If a parameter is not given in the command string, the configured default is used, otherwise the command string overwrites the default - Editor]*

Anyway, the upgrade was installed - it went in first time using the new instructions - and was fired up for the first time configured exactly as it had arrived. Everything seemed to be in order, but when QPAC2 tried to load, I noticed the first problem.

My boot file on the QXL\_WIN file on the PC's C:\ drive - ie, the QXL's WIN1\_ drive, was attempting to load PTR\_GEN, WMAN and HOTKEY 2 and it appears that these are built in to QPC in SMSQ/E. As QPC detected and used the same 'hard drive' as the QXL had been using, it used the same boot file and got slightly confused. Typing 'CONTINUE' at the prompt got the system up and running. All my normal stuff was loaded - not as much as I have seen in some boot files (Hi Jochen!) but sufficient for my development needs.

Right then, I thought, let's get some work done. At this point my second problem was discovered. No mouse! The mouse on the desk moved around with no problems whatsoever, the pointer on the screen stayed resolutely in one place and seemed to regard moving around on the screen as well beneath itself. Not a major problem but one to drive me mad. Lets check the manual again.

The manual gives details of how to set up a 'special' MSDOS session so that QPC can be run from within Windoze. This allows QPC to take over the entire PC and most of its memory - you tell it how much memory to use - and how to load the mouse driver software. Being a QPC (and to a large extent, Windoze 95) novice, I had blindly typed in exactly what the manual had said. A quick perusal of the commands and a quick 'explore' of the PC's hard drive led me to conclude that I was not loading the mouse driver from where the mouse driver lived!

Back in Windoze, a quick edit and we were loading from the correct place. Reboot QPC and the mouse driver was found and loaded - it even tells you on the screen, if you read it that is! Press any key to continue (or ESC to abandon) and we were back into QPC and problem 1 arose again. 'CONTINUE' was typed, and off we go, QPAC loaded along with all the other stuff and problem 2 arose again - still no mouse!

Back to MSDOS and we do the usual command 'MOUSE /?' to get help on how the mouse works. It can be configured for lots of different settings - so, still in MSDOS, I try a few. All to no avail, the mouse will not work in MSDOS mode at all, so it is hardly surprising that it doesn't work in QPC. No problem, I just use the cursor keys. I still have not got the mouse working outside Windoze.

At this point, where I have decided to continue without mouse support, I encounter - or rather don't encounter - problem 3. The manual clearly states 'Do not install QPC onto a compressed PC hard disc' - oops, slight problem here, all my system discs are compressed - Windoze and its programs take up far too much room and also the 16bit FAT (File allocation table) in MSDOS causes my large drive to use 32K clusters (basically every file is a minimum of 32K long - even if it contains a single character. I know how to fix it but I can't be bothered. It is far too much like hard work). QPC works fine on my system even though it is compressed. It does help if you read the manual properly before starting to work. I read it once, then again and still missed the warning about compression - not the fault of the manual however .

So I now have a working system, albeit without a mouse, but I can live with this for now. Back in Windoze, I have an icon set up to run QPC so I double click on it and the PC comes up with a little message that says 'I am about to close down Windoze and restart in MSDOS mode, is this ok?'. I click OK and away it goes. Now Windoze 95 is not what you would call the fastest of operating systems to load up, in fact on my system (Cyrix 585 running 133Mhz with 16 meg of memory) it is utterly slothful and seems to load twice before it is ready for use, however, it starts up and loads into QPC exactly as required - magic. (Still no mouse!)

QPC is not at all like QXL as it takes over the whole machine, processor, memory etc. QXL is a separate card that plugs in to the back of the machine and when you switch to it, it uses the keyboard and drives but in a way that can be shared with Windoze as well. With QXL you can switch from the PC to QXL and back again using ALT and TAB (only from Windows) once you have set up a PIF file (Windoze 3.x) or configured the icon properties (Windoze 95).

QPC takes the amount of memory it was given and trashes anything that is running at the time. The PC thought that it was in control - hah, QPC took it by the throat and booted it out the door - QPC now runs the machine and everything in it. This is why QPC cannot be run alongside any other programs - the PC doesn't know that it is there anymore and has totally lost control. QPC has taken the memory over using its own memory management routines and if anything else was to be running then who knows what would happen.

Having said that, you can switch back to

MSDOS and carry out rudimentary commands, but not too many of them. The manual gives the details on what to do.

At this point I was in total possession of a hard drive with a few files on it but not my entire system. When I moved house last year I sold my old PC and copied off all of my needed files onto a massive ZIP file on 26 DSHD floppy discs. When I arrived in my new abode, I bought a new(ish) PC and eventually installed my software backups from all those discs, well, I tried to. Disc 23 out of 26 was of course, corrupt and could not be read. Guess which file was on disc 23? QXL.ZIP which was my zipped up copy of QXL.WIN - 'oh dear, how unfortunate' I exclaimed - or something like that anyway.

So, there I was, no QL as it was in storage, no system software apart from a few of my floppy discs and no possibility of rescuing the data from the corrupt disc - what could I do. I had tried every disc rescue program on the internet and on any public domain software and even tried a disc sector editor that I had to see if that could read the disc, but to no avail. It was, as Monty Python never said, 'a dead disc'.

After a bit of thought I found my most recent WinBack backups (oh look, a software plug) but another problem was then encountered - have you ever noticed that things never seem to get any better? The backups were all on QL ED discs and of course, the 'leading edge' technology that the PC claims to be can't read ED discs as it can only manage HDs. What to do?

Off to the storage place, pick up a box of QL bits and pieces, take it home to small flat, set it up on the bedroom floor and format hundreds of HD floppies ready for use. Copy the dumps from ED to lots of HDs in the bedroom and then as each one is copied, run through to the living room and copy onto a newly formatted WIN2\_... back to the bedroom for the next one and so on. Eventually, I had almost got back to normal, but of course, my backups were slightly out of date and most of the program files were missing as I had the original discs and didn't need to back the programs up - oops.

More floppy hunting ensued and eventually most of the system was back up to date, or at least, as up to date as it was going to get! And this is the state I am still in.

Over the next few weeks I proceeded to set up win1\_ as I thought I needed it and copied stuff back over from win2\_ into the new directories, reconfigured to suit and suddenly, I was back in business, almost.

Using QPC has been a bit of a joy, it has to be said. QPC is a great piece of software and probably the best way to get a QL system running under Windoze 95. I say this not to cause any offence to those of you who are using QXL but I have found a real performance hit when using QXL under Windoze 95 - it runs so S...L...O...W...L...Y and it was never a problem under good old Windoze 3.1. It has actually become unusable as it is so slow to use. I have tried upping all the assorted settings and the old PIF file from my 3.11 days was converted to 95 format when I loaded the software but still no joy. As far as I can see, I have done everything I can to get performance out of it but nothing so far - anybody got their QXL running properly under 95, and if so, let us all know through this magazine. (The problem is in SuperBasic's editor - it seems to run ok otherwise.)

Using QPC is just like using the good old QL with a Gold Card installed, at least. All the old favourites are there, in fact, all the necessities are there Toolkit 2, Pointer Environment and with SMSQ/E you get all this and extras. All the new facilities and commands that are available on QXL are here now in QPC too. SBASIC is there with hex numbers, binary numbers PEEKing into the system and Basic variables etc etc - see previous reviews in IQLR and QL Review for full details, there are too many to list.

I am currently in a new and very busy job working in a software house in Leeds. I don't get much time these days to 'play' on my QL and certainly not after a long arduous day in the office, because the last thing you really want to do at night is play on a PC running anything, never mind QPC, if you have been staring at a screen all day. Having said that when I do get the chance to play, it is with QPC and I am using it to get all my old software, that used to be sold by Dilwyn Jones Computing, up to date and improved. I also have some new ideas for QL software that I don't think has been seen before, but that is all in the future.

At present the old favourite DJToolkit is being revamped and converted so that it is easier for me to make changes. It used to consist of a single long - over 120K - source file and it is being converted from that into a number of smaller separate source files that can be worked on independently. This task is being undertaken, using various editors - QD7, Digital Precision's EDITOR & QED - Quanta's assembler package - QMAC & QLINK and even, dare I say it, some PC software.

I am rewriting the manuals using a 'proper' PC package, not because the QL doesn't have anything suitable, but because I use the same software at work and already know how to use it to its best effect for producing user documentation. It all takes time you know, and being a newly wed [I tried to warn him, - editor!], time is not something I have a lot of these days. Eventually, QPC will reign supreme and my software will reappear, until then, QPC is performing correctly and at a good speed. I don't get frustrated with it, nor does it crash causing me to lose my work. So it appears that I will continue to use QPC until further notice.

I will close this article, by saying this. QPC is probably the best program available for the PC. It provides the PC with a decent quality operating system that works, can multitask - which is something the PC's own operating software is just getting to grips with - it runs all my old software with no problems as yet, and it runs them at a decent and workable speed. Floppy disc access is perhaps slower than with my old QL and its ED drives but then again, PC floppy disc access is somewhat slow anyway, and if you have slow hardware, no amount of software will make it go any faster.

The technical stuff behind QPC is not too difficult to understand. The manual explains that the emulator can perform every instruction of the 68000 instruction set although it advises any software that asks it, that it is actually a 68010 processor. This is to stop C programs that use 'advanced' features of the 68020, if detected, to not use them, the cache, for example.

The QPC comes with a CD player program. Written in Basic, this program allows the PC's CD ROM drive to be used to play audio CDs from within QPC. I have not used this program yet as I have a good enough Stereo system - unfortunately still in storage but I use a small portable instead. There are a couple of hard disc utility programs and some patch files for various programs that may have problems running under SMSQ or SMSQ/E. The patches supplied are for :

QREF - to allow it to run on programs written in SBASIC which use binary or hex numbers %10010001 or \$4AFB for example. This is not a QPC problem, it is a QREF problem as it is older than these facilities.

QLIB - to allow it to use EXTERNALS properly from SBASIC jobs that are not job zero.

ARCHRTM - to allow the Archive Runtime

system to detect the constant RTMEM%. This is apparently a problem under QDOS and SMSQ.

C68 - it seems that C68 programs won't run on any SMSQ or SMSQ/E system where the 'processor' command returns a value of 40.

I think that writing this program has to be the best thing that the author has done. He, Marcel Kilgus, has done all QL owners who have a PC, a great service and he should be rewarded greatly for his efforts. If only I had his programming knowledge!

If, after reading all of this review, you fancy getting hold of a copy of QPC then do so. Bear in mind that QPC must be run on a 486 or greater processor and will detect if the processor in your machine is up to the job as part of its startup routine. QPC will not run at all on a 386 as it uses the enhanced instruction set on the 486 processor.

Don't forget, your QPC software must be installed and uninstalled using the master disc. No copies can be made - at least, none that will work - and you cannot move or rename the QPC directory after you have installed it or it won't work and you will have lost one of your two 'installations'. I don't know how floppy problems will be catered for by Jochen etc if you end up with a faulty disc at some point in the future, especially in cases where you have not actually installed the software onto the PC's hard disc, but are running it from floppy all the time. At some point the floppy disc will wear out, and you could be in trouble. I think it is best to install QPC once onto your hard drive and leave it there forever.

***[By the time you read this, QPC V1.30 will be released. The version after Norman's version offered the usage of SER1 to SER4, and the floppy disk access was as fast as on a QL. V1.30 will add more speed to QPC in general, my test version ran between 10% and 20% faster!]***

***The mouse: first of all, it has to work under DOS, of course! If it works under DOS, then the only thing you need to do is: configure the COM-port, to which the mouse is connected, in SMSQE.BIN to "none" - SMSQ/E must not use this port if it is already in use by the mouse!]***

■

## ProWesS Delivers - Part 2

James Hunkins

### Programming

The ProWesS system introduces many of the modern programming techniques found on other platforms. People who have tried to program for the pointer environment on the QL have for the most part found it to be difficult and tedious (some of us think of it as a challenge and welcome it, while others I know of have just given up). Through the use of new functions (similar to the APIs found in other operating systems) in 'C' and SBasic, ProWesS has greatly simplified using a pointer environment and added direct access to powerful graphics and fonts. It allows for faster prototyping and re-entrant code. ProWesS is event driven. The programmer describes the look of the window and then passes control to ProWesS, which proceeds to draw the window. ProWesS then waits for an event to happen (keyboard entry, mouse movement, etc) and then sends a message back to the programmer's code. The programmer then handles the event, modifying windows or doing some other activity as needed and then sends control back to ProWesS. When everything is done with, ProWesS takes care of the removal of the window, etc. Through the use of powerful system calls, the programmer will find that most of his work is in implementing his own original part of the program, not in designing or working with the user interface. This leads to faster development and a consistent look and feel to programs written for ProWesS. One other advantage occurs with ProWesS, screen and device independence. This means that the programmer does not have to worry about the screen size or how their program will look on any display devices. ProWesS takes care of the screen display and printer interface. Again, less for you, the programmer, to worry about. As for programming languages, if you have programmed for the pointer environment, you originally had to program only in assembly. Later 'C' was added thanks to C68 libraries. And eventually SuperBasic was added through other packages you can buy. ProWesS was originally only available in 'C' through the Syslib and PROforma access to C68. For 'C' programmers this has been attractive, again due to the powerful routines supplied. With this release of ProWesS the

user of SMSQ/E can now program using the beloved SBasic. Please note that for SBasic programming, you must be using SMSQ/E, as the interface will not work with the original SuperBasic. However, if the SBasic is compiled, the compiled version will run on non-SMSQ/E machines. In a follow-up article, I will go into more detail about programming in both languages for ProWesS and try to give some examples and illustrate just how easy it is. For now, be assured that the programming documentation for ProWesS is rather extensive. Examples are also available. This is a vast improvement over the case of trying to program with 'C' for the original pointer environment.

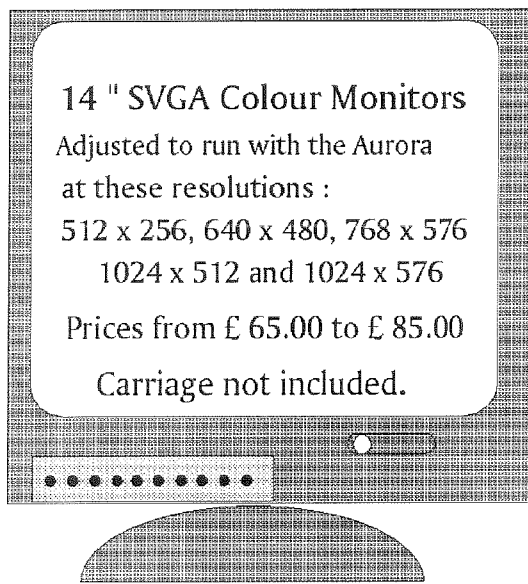
### Performance/Limitations

The old saying that you can't get something for nothing does apply to this case. While ProWesS supplies a lot of sophisticated features and capabilities, it does require a significant amount of CPU power to do so. The speed of the package has been improved constantly since the early versions, but you still should not try to run this on your original QL. I have been running my copy on both my QXL (the 20MHz version) and on a Pentium 100 MHz laptop with QPC. I have also seen it running smoothly on a Super Gold card and believe that it should be okay on a standard Gold Card setup. If you try to run it on a laptop with power saving circuitry, watch your processor settings. My laptop slows the clock speed on the processor when it is running off the battery. So I either restrict it to AC line use or set the processor to full speed while on batteries (the other power saving settings don't have a major impact on it, such as screen shutoff and disk drive slow down). The smaller fonts have been improved on my laptop but I still get some quality problems. Most are however now acceptable. One of the improvements that is discussed for the future in the documentation is the small font quality. By the way, the small fonts on the printer are excellent. Also your display resolution seems to have a major impact on the small font quality. One other suggestion for the included HTML reader comes to mind. Currently you have to use the scroll bar to browse through a document that doesn't fit your screen. An option to use keys for individual line and page scrolling would be a welcome inclusion.

### Documentation

One of the major problems with just about any software (not just in the QL world) is in the documentation. The documentation is either too limited, too involved, poorly organized, or just a plain mess. But here I think that we can safely say that we have an exception. The ProWesS package comes with a fairly detailed but well organized group of documentation. Please note that if you are planning to program with ProWesS, you must download that separate programming documentation either from one of several bulletin boards or from the ProWesS page on the internet. This was probably done as the programming documentation is much more extensive than the standard package documentation and would only serve to confuse the non-programmer (not to mention take up disk space). Also, being programming documentation, it tends to change more often and this allows the programmer to get updates. In order to make the documentation easy to use, ProWesS has chosen to use the HTML format and supplies the Reader program to view (or print) it. HTML allows the documentation to be nicely formatted and allows links to be included. For example, if you are reading about something and you see a reference to another place in the same or another document, you simply place the cursor over the 'link' (usually underlined) and the reader will load the other document. When you are done reading the new document, you can choose 'Back' and return to the previous document. It makes for a much more pleasant experience (I can't believe that I said pleasant when referring to software documentation!). And since HTML is a widely used standard, you can also read the documentation on just about any web browser on just about any machine. It should be observed that ProWesS chose to use the HTML format for documentation some time ago. Today, other operating systems and software are just now beginning to do the same. In fact it is almost an explosion of effort to convert documentation to HTML. An impressive display of foresight. A tip: the reader documentation suggests that you look for a main menu [sometimes the program's name + the \_html suffix] or table of contents [\_toc\_html] HTML file to start viewing hypertext documents (many documents are made up of several separate files). I have also found that files named intro\_html, introduction\_html or index\_html can also be good starting points.

# Get a better view



## 14 " SVGA Colour Monitors

Adjusted to run with the Aurora

at these resolutions :

512 x 256, 640 x 480, 768 x 576

1024 x 512 and 1024 x 576

Prices from £ 65.00 to £ 85.00

Carriage not included.

**Q Branch**  
 Feeling out on a limb ?  
 Reach out for Q Branch  
 Suppliers of Quality QDOS/SMSQ products  
 Hardware and Software

P.O. Box 7.  
 Portslade  
 E. Sussex  
**BN41 2ND**

Tel : 01273-386030  
 Fax : 01273-381577  
 Email :  
 qbranch@qbranch.demon.co.uk

## Q Branch News

The summer time is never the best of times to sit in front of a computer screen and this summer has been no exception.

Q Branch has, however, been working to acquire a source of small hard disks suitable for use with the Qbide interface. At the moment there are not very many of these but we hope to build up our stocks as time goes by so that we can offer a wider range. Prices vary from £ 25.00 for a 45 Meg model and all disks will be QL formatted ready for use.

We do have a few new programs currently undergoing development and beta testing and we hope to have more details of those in the next issue.

**WATCH THIS SPACE !!**

Miracle systems have formally announced that they are working on their new Ultra Gold Card which will have a 68060 processor and probably be available in two models ( with and without a floating point co - processor ). This will be designed to fit alongside a Super Gold and therefore be a dual processor system. ( See their announcement in this issue ).

The current batch of Super Gold Cards are selling well and remember, this will be the last batch available because some of the parts are obsolete.

**BUY ONE NOW WHILE STOCKS LAST !!**

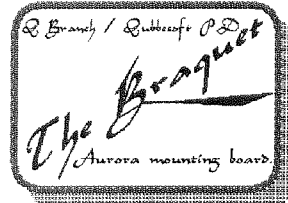
## Hardware

QXL II	£ 200.00
Super Gold Card	£ 160.00
Recycled Gold Card	£ 60.00 *
Aurora	£ 120.00
Qbide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
The 'Braquet'	£ 16.00

\* when available.

## Package deals

QXL II + SMSQ/E	£ 260.00
Aurora + SMSQ/E	£ 180.00
Aurora + SMSQ/E + Super Gold Card	£ 330.00
Super Gold Card + Monitor	£ 380.00



NEW  
PRODUCT !

**£ 16.00**



Payment accepted by Switch, Visa, Mastercard Eurocheque, or cheque drawn on a UK Bank in pounds sterling. Prices include P&P in Europe.



Send a stamped addressed A 5 envelope to us for a complete catalogue of QDOS/ SMSQ/E goodies. Check out our demo disks £ 1.25 each + 50p P&P.



# Q Branch

Feeling out on a limb ?

Reach out for Q Branch  
Suppliers of Quality QDOS/SMSQ products  
Hardware and Software

P.O. Box 7, Portslade, East Sussex.  
BN41 2ND  
tel: 01273-386030 fax: 01273-381577  
Email :  
qbranch@qbranch.demon.co.uk

'Just Words' by Geoff Wicks  
THESAURUS, STYLE CHECK, SOLVITPLUS 2  
£ 15.00 ea / ANY 2 PROGRAMS £ 25.00 / ALL 3 PROGRAMS £ 35.00

**QPC**  
SMSQ/E owners  
£ 78.00  
non SMSQ/E owners  
£ 100.00

## PROGRAMMING

QD v 9.04	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1	£ 37.00
Easyptr pts 2 & 3	£ 18.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

## SMSQ/E

Gold Card / Atari / QXL version  
£ 76.00

Various Atari versions : call for details

!!! NEW PROGRAM !!!

## QCOUNT

Pointer driven home accounts program. Keep track of your bank account, stocks, etc. Stores it's data in DATAdesign format

£ 25.00

## UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread	£ 66.00
Cueshell 2	£ 37.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 28.00
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00
LDUMP	£ 26.00

Q Branch high quality Mouse Mats  
£ 5.00

## ProWesS

ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.14	£ 24.00

## Text 87

£ 79.00

Typset 94 £ 29.00  
Fountext 94 £ 39.00  
2488 drivers £ 29.00  
Epson ESC/P2 drivers  
£ 26.00

Text 87 is the only QDOS /  
SMSQ wordprocessor capable  
of handling the full screen on  
the Aurora / QXL / QPC  
systems. New drivers are  
currently being written.  
WATCH THIS SPACE !

## NOW AVAILABLE !!

Page Designer 3	£ 40.00	Flashback SE	£ 40.00
QIndex	£ 20.00	Flashback	£ 25.00

PLUS MANY OTHER PROGRAMS PREVIOUSLY SOLD BY QUO VADIS - call for details

## Installation

One of the debates that constantly crops its ugly head within the QL community is whether or not we need simpler ways of loading software. The 'old hands' claim that it would just be extra fluff and that any competent QL user should be able to manipulate boot files, pointer buttons, etc. However, there are a substantial number of computer users, not only within the QL community, that simply don't care to spend the time to figure these things out, whatever the reasons. The results are that a large number of users don't really get a chance to use their QL systems to the fullest. Enter another important piece of the ProWesS package, the loader (PRG\_LOADER). This piece of software allows the user to automatically load software. ProWesS comes on a floppy drive that can boot the QL based system and start the installation procedure. The boot program determines whether your system needs the pointer environment or has it included (QXL/SMSQE users). It then loads the extensions it needs, loads the basic ProWesS system onto the QL system, and starts the installation through the ProWesS loader. All this is automatic up to this point and requires no user work other than to insert the floppy and turn on the computer! Considering the complexity of the ProWesS package, you will be amazed by how simple installation will be. The following notes are from a walk through of the installation onto my laptop. They are intended to show you how simple the install has been made. In fact, I think my notes here might be longer than the install process. Upon completion of the bootup, a window automatically opens with the title 'ProWesS - start here'. You have two options, to either display the documentation or to install ProWesS. The choice is made by moving the pointer to your choice, either with a mouse if your system includes one or with the cursor keys. No previous knowledge of the pointer environment is required. If you choose 'display documentation' the ProWesS reader opens, with a dialog box showing the progress of the document being loaded. Depending on the speed of the QL system and the floppy access, the initial load and display of the document can be somewhat slow. This will improve once your system has been configured. But for now, the dialog box showing progress is a welcome touch. As much as I hate to say it, you don't really need to read this documentation right away. The loading is that simple. In fact, due to the current speed of ProWesS, I found myself waiting to read most of the documenta-

tion until everything had been initially installed and configured. An exception to this advice will occur later when you configure your system and access the Help option (more on this at the proper place). The other choice in the menu is to 'install ProWesS'. Simple enough, just move the pointer over this choice and push either mouse button or the 'Enter' key. Another window pops up telling you to insert the floppy disk (you already have) and asking the location of the install floppy ('install from :'). The default is flp1\_. You can either change the source location if needed or accept the location by picking the 'do' option (or Entering over the location flp1\_text). Now it gets fun. The ProWesS installation window opens. The default destination is given in the first space (win1\_pws\_) which you can change if you desire. Then a menu of 8 options is given. The first is 'info about installation'. Unlike the comment about reading the main ProWesS documentation I made earlier, I recommend you take just a few minutes for this one. It gives you a very concise overview of the main installation choices; hard disk, high density disks, or double density disks. In the documentation, by selecting the install type you are interested (notice that these options are underlined, indicating that they are links to somewhere else), the Reader will jump you directly to that section, avoiding the need to scroll through the entire document (an advantage to their using the HTML format for the reader). You will note some other underlined words in each section. They will take you to other relevant notes automatically. And when you are done reading those notes, you can use the 'Back' option of the reader to return to this document. Please note any special items about your installation from this document as you proceed with the following install notes. If this is the first installation of ProWesS, you should now choose the base installation. If you are installing an update, use the update installation which will avoid writing over previous configuration information. [Note to previous users: if you have downloaded your update, you need to copy the new files onto your original installation disk (always your backed up one, not the master copy). Use this updated disk for the installation]. After choosing the base or upgrade installation, ProWesS loader will now copy the files to their proper location, according to your choices. A progress report is shown giving each file name as it is copied. When this step is done, you are given a choice of different screen resolutions (only for the base install, not the upgrade). Your choices

will be 512x256 or 800x600 in 4 color or 640x480 mono. These choices simply copy a configuration file with suggested values for the resolution you picked. You are free to change any of these values for your personal preference at a later time. In case you are curious and want to look at the original choices for all three options, they are found on the floppy in the `install_skel_` directory with the last letters in their names as `_cfg`. I found their recommendations were fairly well done and only did some minor tweaking later (I hate not being able to play around with stuff, so I always do it whether I need to or not!). The next choices are to install the documentation (YES, pick this. The documentation is worth it) and to install the printer drivers (assuming of course that you eventually want to print something). The documentation will install with the standard progress window. The printer has a few more options and actually uses the `PFconfig` program (you can always install or reconfigure these drivers later). When you choose to Add a driver, it brings up a selection for both printer and special screen drivers. Note the help option in this window. It will again bring up the ProWesS reader with a help file discussing the different drivers and their configurations. You might want to keep this open during configuration. You will want to install at least one printer driver [choices include QVME, HP LaserJet and DeskJets, Epson 9/24 pin and stylus color, NecP5, and the Canon BJ10]. You also will be given several other choices including a fax driver, picture drivers, and bitmap drivers (used by LINEdesign and other programs). You are allowed to mark several choices at once and then use the DO option to install them in one single run. Again, please refer to the brief but useful documentation to help you make your choice. And remember, it is always easy to add other drivers in the future. As for the configuration of the printer driver and the Memory options, I will wait for the next section before I discuss them. I found that I wanted to get everything up and try it before I started playing around with this stuff. The defaults seemed to work without my messing with them for now. Before leaving this section, you can choose your Default printer selection (if you have more than one printer model or resolution). Note that even though you might have only chosen a single printer, some of the drivers have multiple resolutions, so you might still want to choose a default here. For example the Epson Color Stylus loaded both 360 and 180 dpi options. Also, load your Fonts here. This choice will list all

the fonts in your floppy drive (you can load others later with multiple paths, etc - please see your documentation for details). Again, you can mark all the ones you want together and use the DO option to load them all at once. The Search directory for fonts option refers to loading other fonts from different directories and can be safely ignored for now. Make sure you hit the 'Make changes permanent' option before leaving this section. Finally, you are almost done. Before continuing, I recommend that you make a copy of your original boot file. The installation will make a recommended boot file that loads all the required items to make ProWesS work. When you choose this option, you are asked what file location should be used. Normally you should write it to a different location other than your normal boot file root directory so that you can merge it into your original file. The reason that I recommend you backup your original file before this step is that you, if you are like so many others, may be distracted and tell it to use your original boot location. This particular installation will be happy to follow your instructions and simply overwrite your original file without a warning. Oops! Of course this will not happen to you, but just in case... When merging the new boot file into my original boot file, just about everything was straightforward. If a program is loaded by ProWesS that is not in your boot file, add that line. If you already load that program, be sure that you have a current version (ProWesS comes with the required files and should have fairly current ones). For SMSQ/E users, be sure to pay attention to the comments about not loading `ptr_gen` and `thing system` (they are already included in SMSQ/E). The only problem I had was with the location in my boot file for the line that runs the `'prg_loader'`. In its original location it had problems starting the loader. However, by moving it towards the end of my boot file after everything that is required by ProWesS had time to be loaded and startup properly, everything worked like a charm. The last two choices apply only to Qpac2 or PROforma fontpack owners. If you have not loaded Qpac2 previously, this option will do it for you (this illustrates that the ProWesS loader can also be used to simplify the installation of other programs, whether they be ProWesS based or not - could this be a hint?). And if you have the PROforma fontpack, this last option will install it for you. At this point, ProWesS is loaded and is basically configured. You can either reboot your QL system from the boot file that the loader made for you or merge the boot file into your

original. Once this is done, try rebooting your system. And welcome to the a new way of doing software on our QLs.

## Optimization & Configuration

There are three types of configuration that involve the ProWesS package. Configuration of ProWesS itself affects the way the ProWesS windows are displayed and affects all ProWesS applications. Configuration of PROforma involves the printer drivers, the fonts available for use, and the memory requirements (which is more of an optimization for your system). You also may be able to configure the individual ProWesS based applications beyond the global configurations just mentioned. This may be done within the program giving additional control or even overriding global configurations. It may also occur through the use of Qjump format config blocks within the individual program (same as is commonly done in the pointer environment). It should be noted that ProWesS is designed to provide as much global configuration as possible. Therefore much of the configuration that was needed under the standard pointer environment programs has already been handled and is not necessary for individual programs.

ProWesS : Both ProWesS and PROforma use configuration files ending with `_cfg` located in the `PWS_MINE_` directory. The ProWesS file is

`'ProWesS_cfg'`.

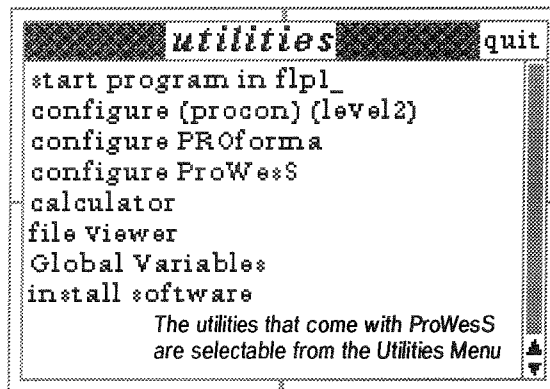
Being a text file it can be edited with your favorite text editor or you can use `PWconfig`, the ProWesS Config program available from the utilities button.

When you installed ProWesS you were asked what screen resolution you wanted to run at. According to your answer a specific version of the `ProWesS_cfg` file was copied from your floppy disk. The settings in this file are ones found pleasing to the authors for the resolution you have chosen. The main difference between the versions are font size, border width, and in the case of the monochrome display choice, difference in colors used. If you are curious you can view the different files on the floppy to see the differences in recommended settings. The `ProWesS_cfg` file holds just a

few types of configuration data, which are either well commented or self explanatory (ProWesS makes good use of real meaningful names). The items of immediate interest to most users involve window item sizes and colors (borders, shadows, backgrounds, ink, paper, fonts). The user can also control default fonts and control the move/drag display methods. For maximum speed, the `SYSTEM_PREVIEW_MOVE` and `SYSTEM_PREVIEW_SCALE` (resize) parameters would be set to 'false'. If this is done, moving or resizing a window will only use the move/resize icons. If these parameters are set 'true' (default), moving or resizing a window will actually redraw the window border for resizing and redraw the entire window while moving. Since system power varies so much, the user can also choose how often during a move/resize that the window/border is redrawn. The higher the number for the `SYSTEM_PREVIEW_TIMEOUT`, the longer between redraws. This will speed things up but makes the movement less smooth. Most colors are specified in RGB (Red Green Blue) format with three numbers, each from 0 to 100 (100 is maximum), which indicate how much of each primary color is required to make the final color. For example, "100 100 100" is white, "0 0 0" is black, "50 0 0" is medium red, and "30 80 0" would be a yellow with a green tint (red and green give yellow in this system).

This obviously allows for more than the current 4 colors given and will work very well with the new Aurora 256 color capability. Another new concept with ProWesS is that the sizes are usually given for a 'virtual' screen size in points. Regardless of the screen size, the software scales it to a size of 720 by 540 points (virtual screen coordinates). There are many parameters available for configuration if the user wishes

to customize things (but note that the suggested defaults will probably work very well for most people), not all of which are included in the `ProWesS_cfg` file. To get more information of all the parameters available, use the ProWesS Reader to view the help files. By opening the `ProWesS_html` file and choosing the Configuration & customization link, you will be taken to a new file where you can find this information. For details on lower level configuration items you can follow the links in the same file or directly open the `_html` files with matching names. For



example, to see how you can configure the EDLINE type (function), open edline.html.

PROforma : PROforma can be configured the same way by editing the PROforma\_cfg file. Additionally, you can use the configure PROforma option found under the utilities menu button. While many users may never bother to configure ProWesS, they will on the other hand probably play around with this one. Therefore the separate program will make life much easier. When installing ProWesS, you will have already seen this program. It has a title bar with PFconfig in it.

```

help|PFconfig|quit Zzz
      Add driver
      Configure printer driver
      Default printer selection
      Memory option:
      Font:, add from directory
      Search directory for font:
  
```

The PROforma configuration program simplifies setting up the system

You can use it to add and configure printer drivers, set the memory options, and add fonts. You probably already added a printer driver (if not, you can do it from here). By choosing to configure a printer driver, a list of drivers installed opens up. For printers supporting more than one resolution, you should see an entry for each resolution. You can pick the highest resolution and just configure one as the configuration applies to all resolutions of the same printer driver. After a printer has been chosen you get a window

```

help|dconfig - driver config|quit Zzz
Epson Stylus Colour 360dpi
unit      mm      inch      pt
default device PAR
width of page 8.50in
height of page 11.00in
left margin 0.00in
top margin 1.00in
reset page size  A4      A3      letter      legal
                Get      Set      Try
  
```

The printer drivers are easily configured and customized

where you can set the default device (normally pard or ser1hr) and set a page size, either by individual size and offset dimensions or by choosing a sheet size (A4, A5, letter, legal). You can choose the units to be displayed in mm, inches or pt. and when done try your results. The program will print a page with lines at set distances marked on the paper. This makes it very easy to adjust for your individual printer's differences. I noticed with my printer choice, there were default settings that pressing a paper size overrode. If you do this by accident and want to restore the original defaults, use the get option to reload them. Make sure you use the 'Make changes permanent' option before you exit or else your changes will be lost. Once

your printer is configured and set, you can now set how PROforma handles memory usage. Again, the defaults will probably work for most people but if you have limited memory or will be using lots of fonts and/or font sizes, you might consider adjusting these values. The settings control how much memory PROforma can use

to render a page with. More memory may speed things up for complex images but if your system is memory deprived or overloaded with memory hungry apps which are open, you might want to sacrifice some potential speed for more available memory for other apps. Another trade-off found in the memory configuration is the use of caches for fonts and colors. Every font that is displayed on the screen or printed has to be rendered (including

different sizes). The font cache allows PROforma to save fonts already rendered in memory so that it doesn't have to render them each time they are accessed. This has a major impact on speed. If you don't believe it, set the font cache to 0 and try to display a page in the reader. You will turn it back on. But again, the

more memory you can spare the larger the font cache, the more fonts you can hold (but will you really be using that many, think about it before you get carried away!). And remember if you limit your QL system in availa-

ble memory, it can slow its operation drastically, regardless of what ProWesS is doing. Once again, a balancing act. But the average user will probably be happy with the defaults. The other thing you can do from here is add more fonts, just in case you purchased the extra font disks from PROGs or grab some Adobe type 1 fonts and convert them.

Other : Some ProWesS programs will still have individual configurations, in addition to using the global ones just discussed. If they are equipped with the Qjump config block, you can still use the standard Config or MenuConfig programs or you can use the Procon version shipped with this package.

## Other ProWesS Applications

The following is a list of ProWesS commercially available applications that I am aware of. In other words, these exist for your purchase and there may be others (either from ProWesS or independent software developers) not listed here.

**FontUtils** : includes three different utilities

**pfb2pff** : conversion of Adobe type 1 fonts to PROforma

**FontShow** : displays an entire font to either your screen or printer

**FontPreView** : shows a sample of any PROforma font loaded for a quick comparison.

**PFList** : prints out high quality listings to your printer with considerable formatting control.

**PROforma Fontpack** : a collection of 100 high quality Bitstream fonts.

**LINEdesign** : a powerful vector-based drawing program (rumor says keep an eye out for color printer capabilities in the future). This package has recently been updated as definitely takes advantage of the ProWesS windowing system in addition to its use of the PROforma graphics/fonts capabilities

**DATAdesign** : a database package.

And here are a few ideas of what could be good commercial programs for ProWesS. Please keep in mind that you don't need to stick to graphical/font based programs to take advantage of this package. Any program that could use windows will benefit.

**HTML editor (visual)** : would allow creation of HTML documents graphically without having to learn the HTML language.

**Enhanced HTML reader** with graphics display capability.

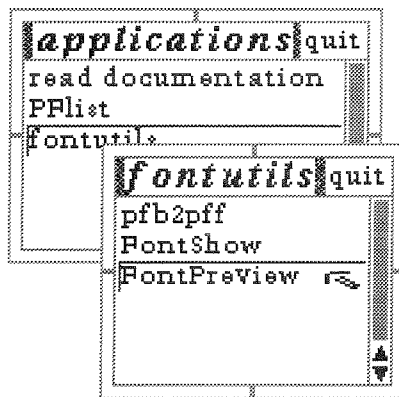
**Greeting card design** (or labels, business cards, etc).

**Business graphics.**

**Desktop publishing** - letter heads (great for holidays)

As I mentioned, with the new ProWesS release and current versions of their utilities, they are designed to be simply and automatically loaded using the PRG\_LOADER program included with ProWesS. It can be found under the Menu Button 'Utilities', 'install software'

option (you may have to scroll the choices down to see this selection). Simply select this option and follow the prompts. Most programs will be loaded from a floppy drive (defaults 'flp1\_') to the destination of your choice. For the three programs I loaded, the installation window where I had a choice to set my destination also gave me additional choices; install with separate button, install in existing menu button, and add documentation to bookmarks file. For the actual installation, you will choose one of the first three choices. The separate button choice will work just like a normal QPAC II button but under ProWesS (the new button will automatically start the program when chosen). The two menu options will add the program to



Automated installation can place ProWesS applications under drop down menu buttons

a menu called up from either an existing menu button already on your screen or a new one (very useful for grouping different types of programs). After you have installed the program, you can use the add documentation option. This adds a pointer to the Bookmarks file used by the ProWesS reader. When you have the reader open, by selecting the Bookmarks option, you can then click on a pointer to this program's documentation and view it on line. All very simple and useful...

## Final Notes - ProWesS

This has been a rather enjoyable article to write. The extensive and well written documentation (just that in itself should excite some of you!) that came with the package has been invaluable and is constantly on my screen. I love exploring new software and new ideas, especially when they are well done. When I first looked at the ProWesS package it was obviously a massive project in an early development state. My first review on the pre-production version showed considerable progress, but there were still speed issues, quite a few bugs, and the promised ease of use had not been added. Now, however, even though ProWesS still has some work to do, I can safely say that it delivers as promised, and possibly more. The ease of installation and use, the customization possibilities, the quality of the fonts and graphics, and the different utility programs included make it a total package. For those of you wondering if you should buy it now or wait... The ProWesS package is now

mature enough to be considered fully functional and useful. If you are planning to buy or update to LINEdesign or DATAdesign, you need to buy ProWesS. If you wish to play with fonts or want to program with windows and mouse pointers or graphics, ProWesS will supply your needs quite well and should make life easier. If you just want to see modern programming practices and keep your system going forward, it might also be the time for you to jump on board. Currently, the number of programs available for ProWesS is limited. But hopefully others will start showing up, especially as the package makes it much easier to do windowed, graphical programs. For myself, along with being a fan of LINEdesign, I will be diving into programming with ProWesS. In fact, I will be delivering a future article or two on the subject, both in 'C' and SBasic. So until then, happy programming. I know I will continue to enjoy playing with this package preparing for the next article.

## IBMDISK Ver 2.00

Darren D. Branagh

Okay, Okay. We all know that QL's are far better than the dreaded Pee-Zees. Just ask Dilwyn! *[PeeZees drive me up the wall much of the time, I don't think I've managed to get one working long enough in one go to ever find out if one machine is better than the other! - Dilwyn]*

However, occasionally we need to transfer files, documents or whatever to the PC format for the benefit of people who don't own QLs. Similarly, a vast collection of very interesting PD text files exist for the PC, and to use these on our trusty QL we need a transfer program. This is where IBMDisk comes in.

Of course, there are many such transfer programs. Some are PD, some shareware, some

Commercial. The more popular ones are XOVER, by Digital Precision, and MULTIDISCover, by Dave Walker. However, both of these cost money - even if the latter is now shareware, and available for a modest fee.

IBMDisk is PD, and available from Steve Johnson, on disk SJP33, which contains version 2.00. It is written by Jonathan Hudson, author of the excellent QTPI communications program. It consists of just one file, called IBMDisk, which you EXEC\_W or EW in the normal way. To describe it as merely a file transfer program is an understatement, as it does much more than that.

On loading, flp1\_ is the default QDOS drive, with flp2\_ being used for DOS and alien formats. The bottom line of the screen displays the current commands available. These are easily selected by moving the cursor keys left or right until the required command is highlighted. The space bar or ENTER key executes the command. These consist of DIR, READ, WRITE, QDIR, DELETE, BOOT, HELP, DEBUG, DUMP, FORMAT, INIT, IGNORE, EXIT.

Many of these will be fairly self explanatory, but heres a brief explanation of the 13 main commands:

DIR displays the directory of an MS-DOS type

```

IBMdisk v2.00
Help is available on the following commands :
DIR   DUMP  DEBUG  BOOT  READ  QDIR
EXIT  HELP  WRITE  INIT  DELETE  FORMAT
IGNORE Hints

For more information on a command, type that command name
followed by ENTER, just press ENTER to leave HELP. If a help
topic takes more than 20 lines, the screen is frozen, press F5
to continue output
HELP> debug?

Dir  Read Write QDir Delete Boot Help Debug Dump Format Init Ignore Exit

```

PC disk in the designated drive (usually flp2). READ reads a DOS file from disk to the QL. WRITE writes a QL file to a DOS disk. QDIR displays the directory of a QDOS type QL disk. DELETE deletes a QL or DOS file. BOOT reads

the Boot sectors of the DOS disk to the QL. DEBUG Prints out a hex dump of given DOS disk blocks. DUMP similar to DEBUG. FORMAT Formats a disk to QDOS or IBM standard. INIT initializes IBM DOS disk. IGNORE Ignores the

familiar QL underscore (\_) automatically by IBMdisk, so these files are copied as FOOBAR\_DAT, CABAL\_LIS, and STABAS\_TXT, to the QDOS medium.

If you ever get stuck, entering the HELP command from the status line and typing "hints" will present a small, well written manual on the screen giving the more general info on the program as I've explained above. It really is very hard to go wrong with IBMdisk - you don't need to be an expert on PCs to use it.

To sum up, IBMdisk is a very well written program, the sort of thing you've come to expect from Jonathan Hudson. In today's computer world, a little

PC compatibility is (almost) essential. IBMdisk addresses this problem - If cash is tight, (and even if it isn't), you can do worse than get hold of IBMdisk, and some PC PD books or text files

disk parameters, and assumes 2 sides, 80 track medium no matter what. This enables the program to use Atari ST TOS disk formats. HELP presents a help page, and by entering each command after the HELP>\_ prompt, displays the relevant information on the command entered, such as its use, syntax, etc. EXIT exits the program.

I found I used most of these regularly, particularly READ and WRITE. In these commands, wildcards similar to those used on "real" PC's are used, with an asterisk (\*) being substituted for a string of characters, and a percentage sign (%) being used for a single one.

For instance, using READ, and entering the file:

\*BA%.\*

as a DOS file, would locate FOOBAR.DAT, CABALLIS, and STABASTXT on the DOS disk. The DOS separator (.) is translated to the

```

IBMdisk v2.0                               (c) Jonathan Hudson 1989-1992
To: field defaulting to the main screen window. The standard QL editing
keys may be used on these fields

Block number fields (Start and End), take numeric values 0 - 1439

The 'EOL Xlate' and 'Query' fields are toggle fields with values 'Yes' or
'No'; the values are toggled using left and right arrow keys or space bar
The Action field is a toggle field with values 'Do' or 'Abort'; select
'Abort' if you don't want to do the action you've selected.

To 'Do' or 'Abort' the operation, select the 'Return' field and press
ENTER, or press escape when in a 'toggle' field (or edit field with) the
wonderful MINERVA ROM. Move through the fields using the up (↑) or
down (↓) arrows

To get HELP on HELP, i.e. to see the commands available type HELP
in response to the HELP> prompt. Type the command name in response to
HELP> to get help on any topic. If still stuck, read the manual !
Type ENTER to return to the command menu
HELP> 3
Dir  Read Write @Dir Delete Boot Help Debug Dump Format Init Ignore Exit
  
```

```

IBMdisk v2.0                               (c) Jonathan Hudson 1989-1992
DOS Directory *.*
IO      .S From      : *%
MSDOS  .S To        :
BOOT DIS.K EOL Xlate : yes      Query   : No
COMMAND .CO      : OK          Quit
DBLSPACE.B
ATTRIB .EX
CHKDSK .EXE 20 6:20:1 22/ 9/1993 0000 12241
DEBUG  .EXE 20 6:20:1 22/ 9/1993 00DC 15718
EXPAND .EXE 20 6:20:1 22/ 9/1993 00EC 16129
FDISK  .EXE 20 6:20:1 22/ 9/1993 00FC 29334
FORMAT .COM 20 6:20:1 22/ 9/1993 0119 22916
MEM    .EXE 20 6:20:1 22/ 9/1993 0130 32502
SYS    .COM 20 6:20:1 22/ 9/1993 0150 9432
QBASIC .HL_ 20 6:20:1 22/ 9/1993 015A 113105
QBASIC .EXE 20 6:20:1 22/ 9/1993 0201 194309
Bytes used 654463

Dir  Read Write @Dir Delete Boot Help Debug Dump Format Init Ignore Exit
  
```

of which there are thousands - and some of which are quite excellent - not a bad way to get a free read, is it?



# JOGHNEW MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany

☎ 0203-502011 Fax 0203-502012 Mailbox 0203-502013 & 502014

EMAIL: SMSQ@J-M-S.COM http://www.J-M-S.com/SMSQ/

## "QL" on foreign Hardware with SMSQ/E V2.85

It is possible to run the operating system of the QL (or the much better one: SMSQ/E) on nearly every other hardware. The idea is to make SMSQ/E available for most hardware platforms, so that programmers and users can benefit from SMSQ/E's new features.

### ATARI Mega STE and TT

**QVME** - VME bus graphics card which is easily plugged into the VME-slot of the Mega STE or TT. No soldering! Very flexible graphics card which can be programmed to QL MODE 4 resolutions of up to 1280x700 pixels and higher. Especially the TT is turned into a very powerful QL system which makes use of the full hardware of the Mega STE and TT (up to 4 serial ports, PAR port, ACSI and SCSI harddisks etc.).

**QVME + SMSQ/E bundled for only DM 599,-**

### All other ATARIs

SMSQ/E can run on all ATARI 520, 1040, ST, STE, Mega STE, TT (but not the Falcon) without any extra hardware required - all you need is SMSQ/E for an ATARI with the extra Monochrome-Screen-Driver.

### Hardware-Emulator for PC

**QXL2** - the new hardware emulator for PCs. A 68040 processor running at 25 MHz makes sure it is running extremely fast. The card has to be plugged into an ISA slot in your PC. With QL network ports. Provides screen resolutions of up to 800x600 pixels.

**QXL2 + SMSQ/E bundled for only DM 769,-**

**QXL2 without SMSQ/E for only DM 619,-**

### Software-Emulator for PC

**QPC** - is ready and works extremely well. See reviews in QL Today (very detailed review in issue 4, volume 1 and current issue). More on page 59!

## SMSQ/E for existing Systems V2.85

**SMSQ/E** is the new operating system which allows you to run your QL programs and adds an enormous amount of additional features: faster, flexible disk format, multiple and much faster BASICs, faster screen driver and much more!

**For QXL & QXL 2**

**DM 199,-**

**For ATARIs with QL-Emulator**

**DM 199,-**

**For ATARIs without QL-Emulator**

**DM 249,-**

**For GoldCard & SuperGoldCard**

**DM 199,-**

## Software, Games, Applications & QL Spares

### QL Games

BlackKnight Chess . . . . . DM 119,90  
Pipes . . . . . DM 29,90  
BrainSmasher . . . . . DM 39,90  
Arcanoid . . . . . DM 39,90  
Firebirds . . . . . DM 39,90  
QShang . . . . . DM 39,90  
Diamonds . . . . . DM 39,90  
The Oracle . . . . . DM 39,90  
MineField . . . . . DM 39,90  
Double Block . . . . . DM 39,90  
The Lonely Joker 2 . . . . . DM 59,00  
SuperGamesPack . . . . . DM 90,00

### QL Spares

ZX8301 . . . . . DM 19,90  
ZX8302 . . . . . DM 14,90  
Keyboard membrane . . . . . DM 28,00

An ideal time to get things NOW - the £ is worth DM 3,- (!!!) at the time I write this!

### QL Applications

**QD Editor** . . . . . [V9.12] . . . . . DM 125,00  
**QD Upgrade from V8** . . . . . DM 24,90  
**FiFi II File Finder** . . . . . [V4.12] . . . . . DM 49,90  
**FiFi II Upgrade from previous Version** DM 19,90  
**QMAKE** . . . . . [V4.16] . . . . . DM 44,90  
**QLiberator SuperBASIC Compiler** . . . . . DM 139,00  
**QLoad-Ref** . . . . . DM 49,90  
**QLQ** . . . . . [V1.13] . . . . . DM 69,90  
**QMAC Macro Assembler** . . . . . [V1.01] . . . . . DM 69,00  
**QMENU** . . . . . [V7.03] . . . . . DM 41,90  
**QMENU Upgrade** . . . . . DM 16,90  
**QPAC 1** . . . . . [V1.05] . . . . . DM 61,50  
**QPAC 2** . . . . . [V1.38] . . . . . DM 119,00  
**QYYP 2 Spell-Checker** . . . . . [V2.17] . . . . . DM 82,50  
**QPTR Pointer Toolkit** . . . . . [V0.29] . . . . . DM 89,90  
**QSpread Spreadsheet** . . . . . [V1.42] . . . . . DM 169,00  
**QSpread Update from V1.34 or before** . . . . . DM 16,00  
**QSpread Update from V1.35-1.41** . . . . . free  
**QSUP** . . . . . [V3.08] . . . . . DM 79,90  
**EPROM Manager** . . . . . [V3.01] . . . . . DM 61,50  
**WINED** . . . . . [V1.19] . . . . . DM 49,90  
**I/O 2 Toolkit** . . . . . [V2.16] . . . . . DM 99,00  
**BASIC Linker** . . . . . [V1.12] . . . . . DM 49,90

### Applications

**LDUMP** . . . . . [V1.05] . . . . . DM 65,00  
**DISA Interactive Disassembler** [V3.02] . . . . . DM 95,00  
**DISA V3 - Upgrade from V1 or V2** . . . . . DM 35,00  
**EasyPTR Part 1** . . . . . DM 89,00  
**EasyPTR Part 2** . . . . . DM 49,00  
**EasyPTR Part 3** . . . . . DM 49,00  
**Stylus-Driver for text87 and text91** . . . . . DM 69,00  
**HyperHelp for BASIC** . . . . . DM 44,90  
**DiskMate 5** . . . . . DM 69,00  
**CueShell** . . . . . DM 95,00  
**QDOS/SMS Reference Manual** . . . . . DM 84,90  
**Update sheets from March 1997** . . . . . DM 13,00

### ProWesS + Applications

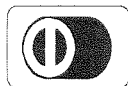
(all ProWesS Applications require ProWesS which is not included!)  
**ProWesS WindowManager+HTML Reader** . . . . . DM 129,00  
**DataDesign Database** . . . . . DM 79,00  
**LineDesign Vektor/DesktopPublishing** . . . . . DM 79,00  
**fsearch** . . . . . DM 49,00  
**Pflist** . . . . . DM 49,00  
**fontutils** . . . . . DM 79,00

### TERMS OF PAYMENT

Postage and package [Germany] DM 8,80 (if total value of goods is up to DM 50,- then only DM 5,80). [Europe] DM 14,- (if total value of goods is up to DM 50,- then only DM 9,-). [Overseas] between DM 14,- (1 item) and DM 35,- (maximum). All prices incl.



15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.



# Multi-Processing, what is it good for? - Part 2

Nasta (Zeljko Nastasic)

## But enough about problems...

Lets talk about solutions. I have attempted to outline most problems that have to be tackled in order to make an efficient MP computer. Now, let's construct a hypothetical machine. What do we do?

First, let's explore the obvious applications of MP, separating the 'processing' aspect of a computer from the data input, output and storage (or for short, the interfacing) aspect of it. The latter we delegate to a separate subsystem, so the data going in and out of the processing segment is only abstracted to favor the processing segment itself. Such a specialized system might in turn use MP s well. Some examples

- make the input/output units more intelligent. For instance, making communication ports be able to handle complex protocols directly, or making storage devices intelligently cache the data, or do data compression. Take special care with user interfacing

- a separate 'human interface' subsystem can free incredible amounts of processing power, and because it's a fairly specialized affair, you can get away with saving money on universality to gain speed and offer specialized functions. In addition, such an approach keeps sometimes very substantial flows of data local to one subsystem, where they will not interfere with other processing tasks - just imagine how much a single code for a letter A destined to be printed on the screen is expanded, both in matters of the amount of data being processed and in the numbers of instructions executed for that to happen, until it actually gets printed. Even adding a single and relatively slow processor to say handle the graphics can improve things dramatically. Adding more to handle windowing, 3D, image compression and audio can make the difference between a mediocre performer and a professional workstation!

What we are left with what represents the more complex problems, to which there are frankly, no cookbook solutions. However, let's do some exploring:

Parallel execution at the granularity level of a single machine instruction is possible, but to be

able to reduce the overheads of deciding which instructions are to be performed in parallel, the scope has to be a low level one - for instance, by looking at two consecutive instructions in a program, we can examine whether they need the results of each other to execute correctly, and if not, both can be executed at the same time. This concept is frequently used in the later generations of CPUs. The CPUs employing it are called 'superscalar' CPUs, and most of the time they execute more than one instruction at a time, with the great majority being able to execute a maximum of two. Because the scope of decision is so small, all the data needed is localized very close in the system, which makes the necessary hardware possible, if it is located inside a single chip. Superscalar operation can be expanded to more than two instructions, however, we soon have the problem of diminishing returns, because we would need a scope wider than only successive instructions, otherwise, given only successive instructions there are very little cases where the instruction's data requirements don't overlap, making them suitable for parallel execution. It has been shown that for typical CPUs it is completely ineffective to have more than 4 execution units. In fact, adding one more, meaning that 5 instructions can be executed at once, will result in only a few percent speed improvement, because there are very little cases where 5 instructions have non-overlapping data requirements, at least in good quality code.

Parallel execution at the granularity level of a task can be extremely useful in systems where it's normal to have a lot of small tasks running at the same time. Because the tasks are small, the resulting granularity is small as well, meaning that even a simple scheduling algorithm will be able to distribute them efficiently among available CPUs. And, because there is a relatively high number of them, it's unlikely that any CPU will be completely idle. However, what if this occurs? One solution could be a concept of distributed computing, and which has to do with networking. Given an environment of multiple machines, each with a single user, but connected between each other by a network, a mechanism can be used to 'lend' unused resources on one user's machine to another user who's machine is tied in by some sort of a network. This is not as far fetched as it seems in this day and age of massive Internet usage - it would even be conceivable to have thousands of computers loosely connected, sharing resources. Of

course, to use them efficiently, a method should be devised to share only those resources which will not be dramatically affected by the communication overhead of the network. This principle can be widely used - the 'area' of the network can be a single enclosure with many CPU boards, or the whole world using the Internet, or phone lines.

The above solutions are either already being used, or are not that difficult to implement. Some systems are in fact eminently suitable for expanding using these principles - and I'm glad to say that the machines used by the QL fraternity, running QDOS, SMSQ or related system software, fall into this category.

### The more radical approaches

Obviously, there are a lot of "mays" and "coulds" in the above paragraph, and that implies some "ifs" and "buts" as well - arguably, what we need is a system that will exploit the inherent parallelism of the tasks it's running automatically. I frequently call this 'the problem of a multiprocessing text processor', anyone who has the slightest idea of how a word processor program works will be hard pushed to find things in it that can happen in parallel! However, that's exactly what the ideal MP machine should be able to do. Sounds too much like having your cake and eating it too? Anyone that figures out a way to do this at least stands to make a huge amount of money in the coming years, at least I think so. My opinion is that this is the way computer development will go in the next century, and remember, that's less than 3 years from now. Do I have a solution? Well, I'm not a genius, and if I did, I don't think I'd tell you anyway (he he!!!), but that doesn't mean I have no ideas, nor that no one has thought of this before. Many people believe that the answer does not lie so much in hardware as it does in clever software, and tools to produce it. From the above discussions on scope and granularity, it is quite obvious that there isn't a single scope, nor a single level of granularity that will solve the problem of a practical MP computer. Under that, I mean one where the fact that it's a MP machine is obvious only when you add another CPU and everything runs faster.

Many, and myself included, think that a large part of the problem lies in the inability to present possible inherently parallel operations in the most widely used programming languages - after all, you can only say things in a language, for which it has words. However, having a lan-

guage for parallel programming may not solve the problem (in fact there are several languages that have MP oriented versions, which enable the programmer to specify 'co-routines' as opposed to only 'sub-routines'). The reason for this is that leaving it up to the programmer to decide which parts of the code can be executed in parallel mean that the programmer also decided on matters of granularity - and some things that could happen in parallel might not be obvious to the programmer. One can argue that this is not even important - it's the machine that should know what can be done in parallel. This could mean that someone has to write a very very intelligent compiler, and frankly, looking at the later offerings I am very sceptical about this, to say the least. The solution would probably be a language in which the description of a way to accomplish a task never implies any sequentiality at all once the program is 'compiled'. I have put the former in quotes, because it is highly likely that such languages can't be compilers in the true sense - how the actual program will execute will depend greatly on what is already going on in the machine, so for every possible situation the optimal 'compiled' program would have to be different, taking into account the available resources. Hence, there would always have to be an element of 'interpreter' in such a programming language. There are languages that have the property of this 'ambiguous sequentiality' and the fact that they are all interpreters goes in favor of my previous argument. Ideally, a program written using this language should be independent of the hardware it's being run on, and run correctly on a machine with one or a thousand CPUs (not necessarily with the same efficiency - after all, we can't expect it to be capable of miracles).

### Some useful properties

In conclusion, a small exploration on properties of our hypothetical machine which are not obvious, but can be deduced from the above.

1. Fault tolerance. A MP machine working along the lines of the above ideas is inherently much more fault tolerant than the average computer of today - it is highly unlikely that all parts of the machine will fail simultaneously. Remember, it can be highly distributed, according to the distributed computing principle. Also, it is conceivable that in the event of failure, it could continue processing with only partial resources, that is, the ones that didn't fail.

2. Heterogeneous design. If we demand universality of programming, so that programs run on different types of hardware, we have to present an abstraction of it to the programmer and not the actual hardware. Because of this, the actual CPUs and their instructions are masked from the programmer, and in fact, except locally in a single CPU, it is highly unlikely that the programmer will ever work at the granularity level of a single instruction. Because of this, the actual instructions are not that important, meaning the machine could use different kinds of CPUs. In fact, it would even be possible to tailor the 'character' of the machine by including specialized CPUs, which can execute some specialized instructions - like floating point, or matrix operations, for machines that are used for serious number crunching, or hash processors for databases. As long as the machine as a whole can execute all the needed instructions, it will be able to execute the programs. Possibly not with the highest possible efficiency in all cases, but it would work.

3. Non-obvious speed improvements. These would come from two effects

- the ability to exploit non-obvious parallelism in a program, and the fact that context changes occur less often, if at all. Context changes happen whenever the machine has to change what it's doing temporarily. For instance, when changing from task to task in a multitasking environment, or when an interrupt occurs. In fact, context changes occur frequently inside a single thread of execution - temporary data is purged, new temporary data is created, parts of memory are moved out of the way, same variables are used for completely different things, etc. Some simulations have shown that such context changes represent a sizable part of programs, and when they disappear in MP machines, because each context ends up having its own CPU, the apparent speed gain is better than the difference in the number of CPUs executing the task - sometimes quite dramatically. One team that researched this showed that typical algorithms run on a system with 4 CPUs, which should have shown a 4-fold improvement compared to a single CPU, on average show over a 6-fold improvement in speed. However, as the number of CPUs increases, the improvement becomes again proportional to the number of CPUs, provided they all have something to do. Because of this, there is another advantage which is purely technological in nature - whereas a single CPU machine

can be upgraded by a faster CPU, when this occurs with a MP machine it will still be faster than the upgraded single CPU machine, simply because the old, slower CPU is still there, unlike in a single CPU machine, where it was removed in favour of the new one.

### What good is it to me?

I'm sure that you have asked this question more than once while reading this. Well, as a reward to your patience, let me show you a glimpse into the future of the QL.

By now, surely you have heard, or possibly read on the pages of this very magazine, that there are two new products being prepared for the QL, as successors to the SGC. Both will exploit at least some of the above principles. Of course, MP is a huge field, so the advances will not come all at once, but rather in small packages, but they will be there. I cannot be any more specific about the upcoming product from Miracle, but I can about the one I'm designing, and that's the GoldFire. Provisions have been made for the GoldFire to be upgraded by an additional CPU, so that both the concept of intelligent input/output and task distributing is made possible by the hardware. In addition, there will be a provision to use multiple GoldFire boards in a single system.

The QL has always been a very progressive machine, and I for one believe that's one of the major reasons it has survived for so long. Therefore, it's only logical to continue the tradition by introducing MP, and with a bit of luck the QL might have a very bright future...

■

## Snippet's Corner - Part 6

M. Knight

This issue we have a substantial listing for you to type in, but all in a very good cause. This provides a Turbo and Q-Liberator compatible replacement for the QL INPUT command with several important advantages over the normal QL input method especially if you want to write multitasking programs using Turbo or Q-Liberator.

Super\_EDIT\$ is a FuNction which returns a string but it is much more. It allows a default to be set and this can be retrieved with SHIFT/ENTER. The TAB key works as a way of moving through long input lines, as does SHIFT/TAB to move

# Professional & Graphical Software

## ProWesS

ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.

The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands (including pictures) and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.

Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that, when you get a new program, all you have to do is insert the disk and indicate "start the program in flp1", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.

ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

## PFlist

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PFlist can create your listings in two columns and in landscape (or both).

## fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurrences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word delimited string.

## font- utils

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

## DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it.

What's more you can choose to look at only those fields you want, and in any order you specify. And you can select which records you want to view, and which not.

DATAdesign allows you to have some hidden comments for each record, have a general look at the file (in tabulated form) or to transfer a record into the scrap of hotkey buffer, so you can easily import a record in your favorite text processor or editor !

Security is a strong point for DATAdesign. Usually files will be memory based, for maximum speed. Files can also be disk based, making sure all changes are immediately stored on disk, so even in the event of power failure, you can at most loose the changes to one record !

Naturally, DATAdesign is good at sorting and searching. And if you were using another database, you can convert Archive or Flashback files to DATAdesign.

The new v4 of DATAdesign makes the program even easier to use than before. You can now also have QD-style icons on your screen to make the program even easier to operate.

## LINEdesign

This is the program which brought QL computing into the nineties. Finally you get access to a modern technology, "vector graphics." This means that your page will be stored in a mathematical form, and not as a collection of pixels.

With LINEdesign, you can create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can draw lines, curves, circles, ellipses, pies, squares, rectangles, rectangles with rounded corners, and any combination of these to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independant, meaning that the printout will be the same on any printer (e.g. same size and position).

Also LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts can be displayed at any size, rotation, etc. All the fonts which are available to ProWesS can be used in LINEdesign.

LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

**new address !!**

Dr. Fr. Hemerijckdaan 13 /1  
2650 Edegem  
Belgium

tel : +32 (0)3/ 457 84 88  
fax : +32 (0)3/ 458 62 07  
email : joachim@club.innet.be  
www : <http://www.club.innet.be/~year2827>

ProWesS - BEF 2400

DATAdesign - BEF 1200

PFlist - BEF 600

Payment terms :

LINEdesign - BEF 1200

fontutils - BEF 1200

fsearch - BEF 600

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutils and PFlist work (even though DATAdesign uses wman).

All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra costs of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.

If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).

Payment can be done by bureauCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

copies	priority mail			ordinary mail		
	Belgium	Europe	World	Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

backwards. The up and down arrows move to the start and end, pressing F4 forces a redraw and ESCape can be used to abort the edit and return to the calling routine, returning only an ESCape character. This means the program incorporating Super\_EDIT\$ can allow users to abort a save by pressing ESCape if they call up the input prompt accidentally. A maximum length is set for the string supplied so filenames no longer need to be truncated to the QDOS limit after input. Super\_EDIT\$ serves multitasking programs well since pressing F4, the standard QL redraw key, will redraw even if the program is in the middle of input. F5 clears the prompt to allow the user to start from scratch.

It might seem over ambitious to write a replacement for INPUT using SuperBASIC but in fact it works very well on anything other than an original unaccelerated QL interpreter. If compiled with Q-Liberator or Turbo it becomes indistinguishable from built in routines for input as far as speed and responsiveness are concerned. SBASIC also gives the routine a new lease of life.

The Turbo Toolkit includes a similar routine in machine code, but it does not offer the default returning behaviour or ESCape option of Super\_EDIT\$ nor the screen redraw during input. All of these can be important. In addition Super\_EDIT\$ allows the programmer to decide where in the string the cursor is presented, so users can be reminded not to change the file extension for example, or can be given the cursor at the start of input if this seems more appropriate.

## Listing 6b

```
31160 CURDIS#0
31165 CURSEN#Tk_Channel%
31205 CURDIS#Tk_Channel%
31210 CURSEN#0
31250 CURDIS#Tk_Channel%
31255 CURSEN#0
```

## Listing 6

```
100 MODE 4
110 WINDOW 252,202,256,23
120 WINDOW#2;252,202,4,23
130 WINDOW#0;504,32,4,224
140 FOR Chan=0 TO 2
150 INK#Chan;7
160 PAPER#Chan;0
170 BORDER#Chan;1,2
180 CLS#Chan
190 END FOR Chan
200 :
210 Default$="Test_DAT"
220 FileDev$="flp1_"
230 PRINT#2;"Test and demo routine.";
```

The main listing assumes the Turbo Toolkit is present and uses the CURSOR\_ON and CURSOR\_OFF commands. For users without this SuperToolkit II provides the alternative though Toolkit II cannot be distributed with your programs and the runtime Turbo Toolkit can. Non Turbo users should alter the relevant lines to match those in listing 6b.

Once you have typed the test listing run it several times and try various things to get a feel for it. Type some garbage then press SHIFT/ENTER to see the default return to the screen; press F4 to see the screen redraw, use the TAB key and the up and down arrows. Try typing something too long to get a feel for the way the code rejects overlong input. If you like think of some additions and try to implement them, but back it up first.

There is some good news for those who have been reading and don't like typing; the listing of all the Snippet's Corner routines so far will be available from public domain libraries SJPD and QUBBESOFT PD. The file will be bundled with some utilities for programmers to use including the QL's best SuperBASIC listing program (modest aren't I?) and there may be some bonus routines on the disk too. Leave it a couple of weeks to give them time to sort it out then write or phone. For the continuation of Snippet's Corner keep asking for what you want! **[Seriously, write and tell us what you like to see - this was the last of Mark's prepared parts of Snippet's Corner. We also hope to have all the listings on our next cover disk - Editor]**

```

240 PRINT#2;\\\"The default string is:\"\\Default$
250 TestFileName$=Super_EDIT$(1,1,0,\"Enter the filename to save on \"%FileDev$,Default$,36,
LEN(Default$)-3)
260 IF TestFileName$=CHR$(27) THEN
270 PRINT#2;\\\"Escape key pressed, no save\"
280 ELSE
290 PRINT#2;\\\"Save to:\"\\ FileDev$& TestFileName$
300 END IF
310 :
31105 DEFine FuNction Super_EDIT$(Tk_Channel%,Tk_LineNum%,Tk_ColNum%,Tk_Prompt$, Tk_Default$,
Tk_MaxLength%,Tk_CursorPos%)
31110 LOCAL Tk_Choice,Tk_KeyLoop,Tk_Offset%
31115 LOCAL Tk_NewString$(Tk_MaxLength%)
31120 Tk_NewString$=Tk_Default$
31125 Tk_Offset%=Tk_CursorPos%-1
31130 IF Tk_Offset%>LEN(Tk_NewString$) THEN Tk_Offset%=LEN(Tk_NewString$)
31135 IF Tk_Offset%<0 THEN Tk_Offset%=0
31140 CLS#Tk_Channel%
31145 AT#Tk_Channel%;Tk_LineNum%,Tk_ColNum%
31150 PRINT#Tk_Channel%,Tk_Prompt$;
31155 Clear_BUFFER Tk_Channel%
31160 CURSOR_OFF#0
31165 CURSOR_ON#Tk_Channel%
31170 Show_STRING Tk_Channel%,Tk_LineNum%+1,Tk_ColNum%,Tk_NewString$
31175 REPEAT Tk_KeyLoop
31180 AT#Tk_Channel%;Tk_LineNum%+1,Tk_ColNum%+Tk_Offset%
31185 Tk_Choice=CODE(INKEY$(#Tk_Channel%,-1))
31190 SElect ON Tk_Choice
31195 =27
31200 CLS#Tk_Channel%
31205 CURSOR_OFF#Tk_Channel%
31210 CURSOR_ON#0
31215 RETurn CHR$(27)
31220 =254
31225 Tk_NewString$=Tk_Default$
31230 Show_STRING Tk_Channel%,Tk_LineNum%+1,Tk_ColNum%,Tk_NewString$
31235 Tk_Offset%=Tk_CursorPos%-1
31240 =10
31245 CLS#Tk_Channel%
31250 CURSOR_OFF#Tk_Channel%
31255 CURSOR_ON#0
31260 RETurn Tk_NewString$
31265 =9
31270 REMark TABulate
31275 Tk_Offset%=Tk_Offset%+8
31280 =253
31285 REMark SHIFT/TABulate
31290 Tk_Offset%=Tk_Offset%-8
31295 =208
31300 REMark UP
31305 Tk_Offset%=0
31310 =216
31315 REMark DOWN
31320 Tk_Offset%=LEN(Tk_NewString$)
31325 =192
31330 REMark LEFT
31335 Tk_Offset%=Tk_Offset%-(Tk_Offset%>0)
31340 =248
31345 REMark F5
31350 REMark **** Delete whole line
31355 Tk_NewString$=""
31360 Show_STRING Tk_Channel%,Tk_LineNum%+1,Tk_ColNum%,Tk_NewString$
31365 =200
31370 REMark RIGHT
31375 Tk_Offset%=Tk_Offset%+1
31380 =32 TO 191
31385 IF LEN(Tk_NewString$)<Tk_MaxLength% THEN
31390 Tk_NewString$=Add_CHAR$(Tk_NewString$,Tk_Offset%+1,CHR$(Tk_Choice))
31395 AT#Tk_Channel%;Tk_LineNum%+1,Tk_ColNum%+Tk_Offset%
31400 PAN#Tk_Channel%;6,4
31405 PRINT#Tk_Channel%;CHR$(Tk_Choice);
31410 Tk_Offset%=Tk_Offset%+1

```

```

31415     ELSE
31420         Super_BEEP
31425     END IF
31430     =194
31435     REMark CTRL LEFT
31440     IF LEN(Tk_NewString$)>0 THEN
31445         IF Tk_Offset%>0 THEN
31450             Tk_NewString$=Del_CHAR$(Tk_NewString$,Tk_Offset%)
31455             AT#Tk_Channel%;Tk_LineNum%+1,Tk_ColNum%+Tk_Offset%-1
31460             PAN#Tk_Channel%;-6,4
31465             Tk_Offset%=Tk_Offset%-1
31470         END IF
31475     END IF
31480     =202
31485     REMark CTRL RIGHT
31490     Tk_NewString$=Del_CHAR$(Tk_NewString$,Tk_Offset%+1)
31495     AT#Tk_Channel%;Tk_LineNum%+1,Tk_ColNum%+Tk_Offset%
31500     PAN#Tk_Channel%;-6,4
31505     =244
31510     REMark Reset_SCREEN
31515     CLS#Tk_Channel%
31520     AT#Tk_Channel%;Tk_LineNum%,Tk_ColNum%
31525     PRINT#Tk_Channel%;Tk_Prompt$;
31530     Show_STRING Tk_Channel%,Tk_LineNum%+1,Tk_ColNum%,Tk_NewString$
31535     =REMAINDER
31540     NEXT Tk_KeyLoop
31545     END SElect
31550     IF Tk_Offset%>LEN(Tk_NewString$) THEN Tk_Offset%=LEN(Tk_NewString$)
31555     IF Tk_Offset%<0 THEN Tk_Offset%=0
31560     END REpeat Tk_KeyLoop
31565     END DEFine Super_EDIT$
31570 :
31575     DEFine PROCedure Show_STRING(Tk_Channel%,Tk_LineNum%,Tk_ColNum%,Tk_Any$)
31580     AT#Tk_Channel%;Tk_LineNum%,Tk_ColNum%
31585     CLS#Tk_Channel%;3
31590     PRINT#Tk_Channel%;Tk_Any$;
31595     CLS#Tk_Channel%;4
31600     END DEFine Show_STRING
31605 :
31610     DEFine PROCedure Super_BEEP
31615     BEEP 4000,1,10,500,2
31620     END DEFine Super_BEEP
31625 :
31630     DEFine PROCedure Clear_BUFFER(Tk_Channel%)
31635     LOCAL Tk_KeyLoop
31640     REpeat Tk_KeyLoop
31645     IF INKEY$(#Tk_Channel%)="" THEN RETURN
31650     END REpeat Tk_KeyLoop
31655     END DEFine Clear_BUFFER
31660 :
31665     DEFine FuNction Del_CHAR$(Tk_Any$,Tk_DelPos%)
31670     LOCAL Tk_StringLength%
31675     Tk_StringLength%=LEN(Tk_Any$)
31680     IF Tk_StringLength%=0 THEN RETURN ""
31685     IF Tk_DelPos%<1 OR Tk_DelPos%>LEN(Tk_Any$) THEN RETURN Tk_Any$
31690     IF Tk_StringLength%=1 THEN RETURN ""
31695     IF Tk_DelPos%=Tk_StringLength% THEN RETURN Tk_Any$(1 TO Tk_DelPos%-1)
31700     IF Tk_DelPos%=1 THEN RETURN Tk_Any$(2 TO Tk_StringLength%)
31705     RETURN Tk_Any$(1 TO Tk_DelPos%-1)&Tk_Any$(Tk_DelPos%+1 TO Tk_StringLength%)
31710     END DEFine Del_CHAR$
31715 :
31720     DEFine FuNction Add_CHAR$(Tk_Any$,Tk_AddPos%,Tk_AddIn$)
31725     LOCAL Tk_StringLength%
31730     Tk_StringLength%=LEN(Tk_Any$)
31735     IF Tk_AddPos%<1 OR Tk_AddPos%>LEN(Tk_Any$)+1 THEN RETURN Tk_Any$
31740     IF Tk_AddPos%=LEN(Tk_Any$)+1 THEN RETURN Tk_Any$&Tk_AddIn$
31745     IF Tk_AddPos%=1 THEN RETURN Tk_AddIn$&Tk_Any$
31750     RETURN Tk_Any$(1 TO Tk_AddPos%-1)&Tk_AddIn$&Tk_Any$(Tk_AddPos% TO Tk_StringLength%)
31755     END DEFine Add_CHAR$
31760 :

```



Sinclair QL  
PD  
and  
Shareware  
Software

# QUBBESoft P/D

Sinclair QL  
New  
and  
2nd User  
Hardware

38 Brunwin Road, Rayne, Braintree, Essex. CM7 5BU  
Tel 01376 347852 Fax 01376 331267

## PD HD Collection

Over 100mb of  
Public Domain  
and Shareware  
Software on 50  
High Density  
Diskettes. Inc all  
Software from  
current Catalog.  
**£25.00p**

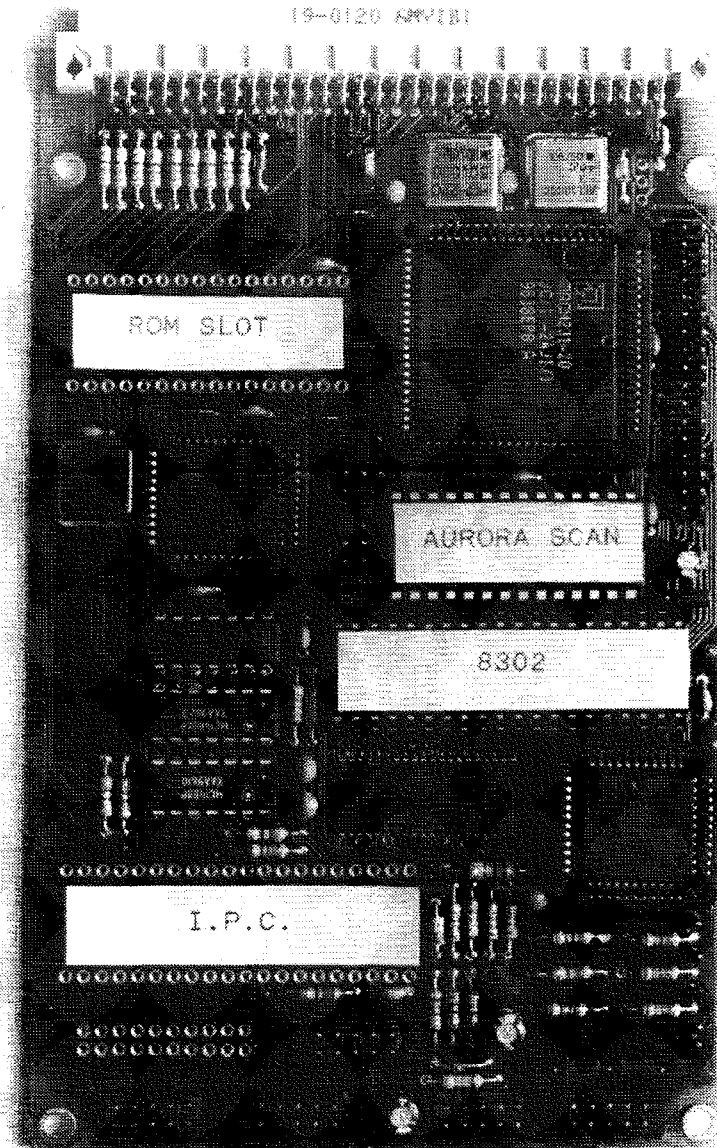
## PD EZ Collection

As above but on  
an EZ135 Cart.  
**£25.00p**

## QUBIDE

Allows AT/IDE  
Hard Disks to  
be connected to  
your QL/Aurora.  
A huge amount  
of storage will  
be available for  
files & programs  
Compatible with  
SGC/GC/TC.

**£55.00p**



## LD Clip-Art HD Collect

Over 100mb of  
Clip-Art files for  
all tastes on 50  
High Density  
Diskettes. Inc  
over 200 \_pff  
Fonts and Cat of  
Thumbnails.  
**£25.00p**

## LD EZ Collection

As above but on  
an EZ135 Cart.  
**£25.00p**

## QPLANE

Back Plane for  
use in placing  
your Aurora or  
QL into PC Mini  
Tower/ Desktop  
Case. 2 Power  
connectors that  
marry up with  
special PC style  
connectors.

**£25.00p**

- Replacement for 13 year old QL Motherboard
- Connect QL/VGA/SVGA or Multisynch Monitors
- 8 Resolutions from 512x256 upto 1024x768
- Super Gold Card and Gold Card (Red) compatible

## Aurora

The QL's Graphics Card

**£120.00p**

Inc A4 Manual and  
VGA/SVGA/MSYNC  
Connector

- Future 16 and 256 colour modes
- Uses SMSQ/E Supported Operating System
- Extended ROM socket (upto 512Kb paged ROM)
- superHermes, Di-Ren Kybd Interface compatible

# Logical Fundamentals -

## Part 1

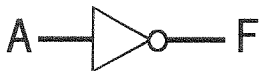
Stuart Honeyball

The inner workings of a computer like the QL are not impenetrable but are a source of fascination. Here I hope to begin to unravel the mysteries of electronic logic circuits that drive computers.

### Boolean Algebra

This is the branch of mathematics, named after a chap called Boole, that describes the functionality of digital circuitry. The terms 'logic' and 'digital' mean the same thing in this context and essentially refer to a system where a variable can have just one of 2 states: true or false. The term 'signal' is widely used to refer to the transfer of information from one place to another i.e. from an output of one device to inputs of others. In electronics 'true' is often represented by a high voltage and 'false' by a low voltage. On paper we usually write '1' or 'H' (high) for true and '0' or 'L' (low) for false.

There are just 3 fundamental elements used to manipulate logic. First, the inverter:

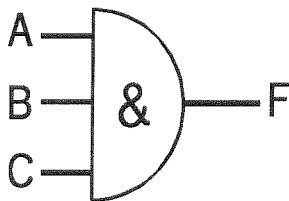


The A is the input and the F is the output. If you apply a high to the input A then F will show a low; and applying a low to A will give a high on F.

This is represented by the Superbasic:

$$F = \text{NOT } A$$

Next there is the AND gate. This combines 2 or more inputs to give a single output. A 3 input AND gate is drawn as follows:

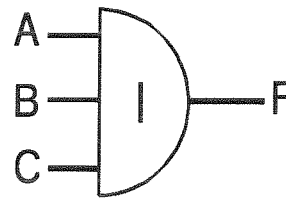


The AND gate is represented by the Superbasic:

$$F = A \text{ AND } B \text{ AND } C$$

F is high when all A, B and C are high, otherwise it is low.

Finally there is the OR gate. This again has 2 or more inputs and a single output:



This is represented by:

$$F = A \text{ OR } B \text{ OR } C$$

F is high when any one or more of A, B or C are high, otherwise it is low.

All the logic in the QL (i.e. all the circuitry bar the I/O and power supply) such as the processor, memories and interface logic can be constructed from these 3 logic elements.

### Truth Tables

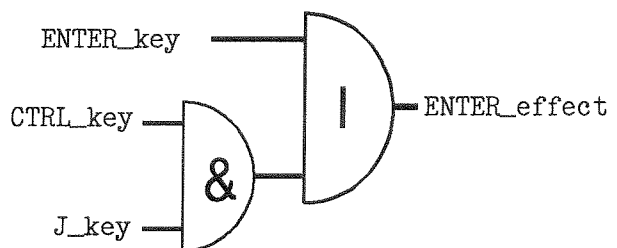
There is a more explicit way of representing the above functions and that is to list all possible input combinations of an element together with the corresponding output levels. For the inverter there are only 2 possible input combinations which makes its truth table quite small:

A	F
0	1
1	0

The following truth tables are for a 2 input AND gate and a 2 input OR gate:

AND Gate:	A	B	F	OR Gate:	A	B	F
	0	0	0		0	0	0
	0	1	0		0	1	1
	1	0	0		1	0	1
	1	1	1		1	1	1

The following example shows how logic elements can be combined. On the QL keyboard the effect of pressing the ENTER key can also be obtained by simultaneously pressing the CTRL and J keys. We could represent this by:

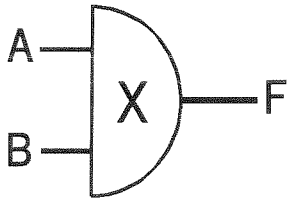


or in Superbasic:

$$\text{ENTER\_effect} = \text{ENTER\_key OR (CTRL\_key AND J\_key)}$$

Suitable names have been given to the input and output signals but note that no name is given to the signal from the AND gate output to the OR gate input because it is not used outside of the circuit. It is worth satisfying yourself that this circuit does the required job.

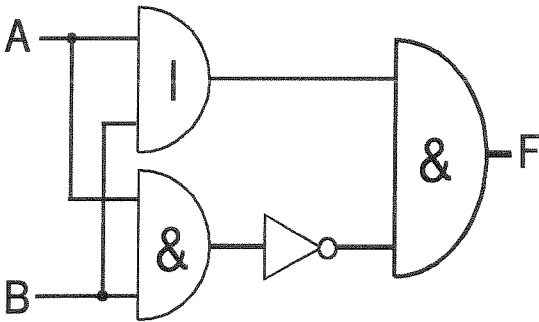
There is another type of gate called the exclusive OR. This is a 2 input device where the output is high when one, but not both, of the inputs are high and is drawn:



Superbasic:

$$F = A \text{ XOR } B$$

It can be made as follows:

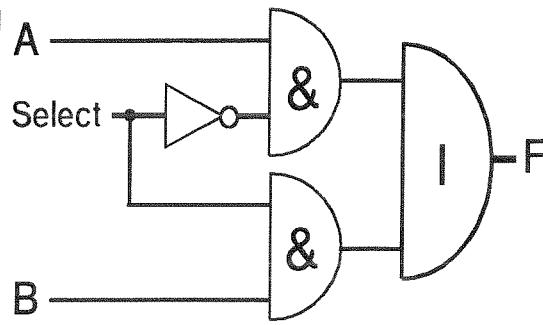


Work through this and you will see that the lower input of the final AND gate is normally high allowing the OR function to follow through except when both A and B are high.

Something reminiscent of the XOR is often found in buildings where 2 switches control one light. If you imagine 1 of the switches to be the A input, the other the B input and the light itself to be the F output, then when you change the state of one switch the state of the light changes exactly like the XOR gate.

XOR Gate:		
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Another widely used circuit is the multiplexer. This selects one of its inputs and passes the value on the selected input straight through to the output. It is used for routing signals and is constructed as follows:



This is equivalent to the following Superbasic function:

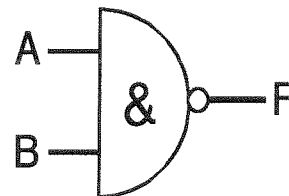
```

100 DEFINE FuNction F(A,B,Selection)
110 IF NOT Selection
120   RETURN A
130 ELSE
140   RETURN B
150 END IF
160 END DEFINE F
  
```

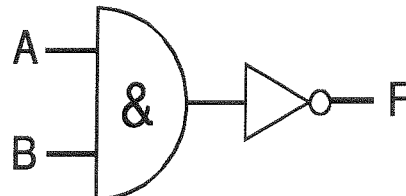
Truth table:			
A	B	Selection	F
0	X	0	0
1	X	0	1
X	0	1	0
X	1	1	1

Note that for convenience the 'X' in the truth table means "don't care" since it can be replaced with either a '1' or a '0' and the table will remain correct.

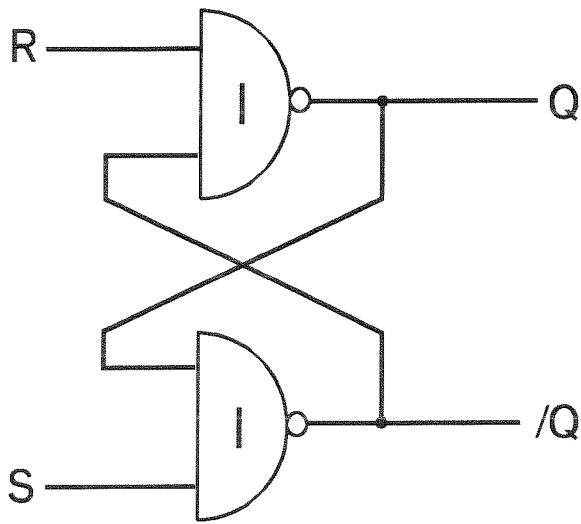
Sometimes a gate followed by an inverter is condensed into a single symbol called a NAND in the case of an AND, and a NOR in the case of an OR.



is equivalent to



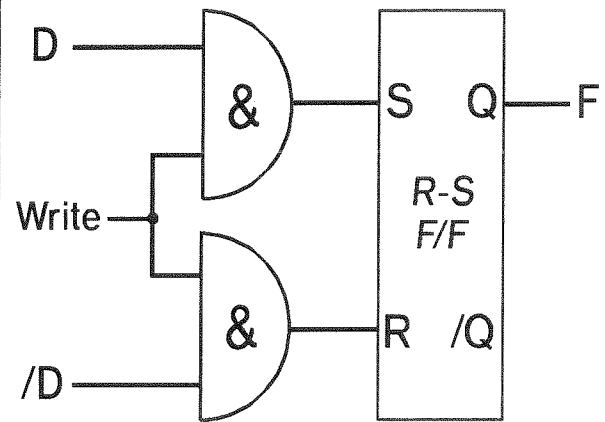
All the above are called combinatorial because they have no storage. In other words the outputs depend purely on the current state of the inputs. Contrast that with the following R-S flip-flop whose output also depends on history:



The R is the Reset input and the S is the Set input. There are 2 outputs Q and /Q (read "NOT Q"). Normally both the R and S inputs are low but look what happens if the R is put high. On a NOR gate if either input is high then the output must be low. In this case the R input is high so the Q output must be low regardless of the value of /Q (its other input). Looking at the lower NOR gate both the Q and S are low so its output, /Q, is high. Now look again at the upper gate; since its input connected to /Q is high its output, Q, must be low regardless of the R input. This means that if R is returned to low the Q remains low and the /Q remains high. Similarly if the S input is temporarily pulsed high (i.e. put high for a while and then put back low again) the /Q will go low and the Q will go high and both outputs will remain in this state. What this means is that the circuit has remembered which of the R and S inputs was last pulsed. The /Q is called the /Q because it is simply the inverse of Q. Usually with R-S flip-flops the state of the

outputs is undefined when both R and S inputs are high and this state of affairs should be avoided although in this case both the Q and /Q outputs will be low temporarily.

The R-S flip-flop can be used to store the value of from a single data input D using the following circuit:



The /D input is derived from the D input by passing it through an inverter (not shown). The box marked "R-S F/F" contains the previously described R-S flip-flop circuit. The circuit operation is quite easy. When the Write input is low both S and R are low and so the F output remains stable. But when Write goes high  $S = D$  and  $R = \text{NOT } D$  so only one of S and R are high. It follows, therefore, that  $F = D$ . Now, when the Write goes low both S and R will be low and so F keeps the state that D was in whilst Write was high. The value that was on D has thus been stored. This is effectively a 1 bit store. If you were to build up 67108864 of these you would have the 8 megabytes of memory found on the QXL!

■

## Label Typewriter

*Dilwyn Jones*

Following on from Darren Branagh's Enveloper program in Vol. 2 Issue 2, here is a simple to use label typewriter program, written in SuperBASIC. Often we don't need to print all of a long list of labels, or we don't need to fire up a large database if all we need to do is to send one or two letters to people whose addresses we can remember from memory.

Quite simply, all this program does is to allow you to type in a few lines of text, then asks you how many copies to print (any reasonable number, in case you wish to keep a supply of

address labels to a particular person, and how much left margin to apply. The margin is simply specified as a number of spaces, meaning you can after a little practice centre the text on the label, for example.

On the assumption that you'll always want to print to the same device, the printer port is assigned in line 130. Change it to `port$ = 'SER1'` if your printer is hooked up to SER1. I use mine on the parallel port PAR on my QXL.

The variable 'lines%' in line 160 assumes you are using continuous labels where the gap from the top of one label to the next is 1.5 inches (38 mm or 9 lines). The value assigned is the number of lines of text between the start of one label and the next. As my printer uses 6 lines of

text per inch, and the labels are 1.5 inches, this means 9 lines of text per label. If you use 2 inch (50.8 mm) labels, you'd set lines% to have a value of 12. Obviously, if there is a gap between the end of one label and the next, you have to leave the last line or two blank when typing in the data. A brief trial and error will quickly show you what you can do with your labels.

On-screen, the display shows a ruler above and below the label typing area. Type in each line, and press ENTER as you would on a typewriter. For blank lines, just press ENTER, of course, with no characters showing. You can move the cursor along a line being entered, and delete characters with CTRL LEFT and CTRL RIGHT arrow keys in the usual way, but you can't move up back to the previous line. If you make a mistake, just break into the program (CTRL SPACE) and start again.

```

100 REMark Simple Label Typewriter
110 :
120 REMark change next line if printer is on a different port e.g.SER1
130 port$ = 'par'
140 :
150 REMark change next line for number of lines per label
160 lines% = 9
170 :
180 REPEAT program
190   CLS : CLS #0 : INK 7
200   CSIZE 2,0 : PRINT' LABEL TYPEWRITER'
210   lines% = 9 : REMark how many lines from top of one label to the next
220   DIM array$(lines%) : INK 2 : CSIZE 0,0
230   PRINT ' .....1.....2.....3.....4'
240   INK 4 : FOR a = 1 TO lines% : PRINT a;':'
250   INK 2 : PRINT ' .....1.....2.....3.....4'
260   INK 7: FOR a = 1 TO lines% : AT a+1,2 : INPUT array$(a)
270   AT lines%+3,0 : INPUT'Print how many copies (0 to quit) ? ';copies%
280   IF copies% = 0 THEN EXIT program
290   AT lines% + 5,0 : INPUT'Left margin - how many spaces ? ';spaces%
300   OPEN #3,port$
310   FOR a = 1 TO copies%
320     AT lines% + 7,0 : PRINT'Printing label ';a
330     FOR b = 1 TO lines% : PRINT #3,FILL$(' ',spaces%);array$(b)
340   END FOR a
350   CLOSE #3
360 END REPEAT program
370 PRINT\'PROGRAM FINISHED.'
```

The ruler has been limited to just 40 characters for two reasons. First, most address labels are about 3 to 4 inches wide, and secondly, if you are using the default monitor windows (e.g. after a WMON 4 command) this will just fit on a line in mode 4.

If you are using a wider window, and want wider rulers, just edit lines 230 and 250, which print the rulers. If using mono monitors, the red ruler ink (INK 2) may be difficult to read, so change the ink colour to 4 or 7 to suit your needs.

This program is neither advanced nor difficult to understand. Having typed it all in, you'll have a quick typewriter ready to prepare instant address labels without having to struggle with a larger database program when it's not really needed.

## London QL Traders Show

*Dilwyn Jones*

For some time, Tony Firshman has organised these shows at the St Helen's Church Hall in London, which is just off the A40 Westway into London. This venue has ample space, is well

known to QL users in the area and has ample on-street parking, making it a popular venue for QL shows.

For those who, like me, got there by public transport, the day was marred by awful delays on the only underground train link to the local station, Ladbroke Grove, caused by "essential engineering works" at Edgware Road station, causing a normally 10-15 minute journey to turn into a 50

minute plus journey with a convoluted change of trains en-route and an infrequent crowded shuttle service for the last part of the journey. Although Tony Firshman is well known as someone who can fix just about any QL problem, even he couldn't sort this one out!

As usual for such shows, most of the QL traders and notables were present. For the first part of the day at least it began to look as if the traders would outnumber the punters, but more and more people attended as the day wore on.

Although not directly selling products at shows these days since appointing QBranch as retail agents, Stuart Honeyball of Miracle Systems was present and keen to promote his latest design, referred to as the Ultra Gold Card. It seems his latest little Miracle will sport a powerful 68060 processor, RAM memory via standard SIMM card slots, feature high quality digital audio capabilities (of interest to me personally, since I'm a sound technician in my day-job) and will require a Super Gold Card alongside it, with interconnection via something like QPlane. Just imagine, a system so powerful it has a Super Gold Card as an I/O processor! At the time of this show, the design was still in its infancy, which means that the alternative Goldfire unit from Qubbesoft P/D should be first to market. Stuart told me that former Miracle employee Jeremy Reeves has rejoined the company to work on new projects. This has unfortunately meant the cancellation of the sound card project detailed in QL Today recently, as Jeremy will now need to concentrate on the advanced sound facilities for the UltraGoldCard.

Qubbesoft P/D's stand was staffed as ever by Ron and Kath Dunnett, and they did quite a trade in remaining stocks of Aurora and Qubide (V1.53 ROM), being sold at a special show price. New features in this version of Qubide allow Discover users to read alien format removable media such as MS-DOS format EZ-135 cartridges. However, if you upgrade to this rom version (upgrades now cost 5 pounds), the map layout has changed, so you'll need to run a conversion utility supplied to update the map. Of course, this demands you make a full backup first, in case something goes wrong! This conversion process needs to be run separately for all removable media. Ron was kept busy fielding enquiries as to the non-appearance of the Aurora high colour drivers.

Qubbesoft also displayed their new issue 25 catalogue, which lists two new general collections of 10 PD/Freeware programs on disks PD15 and PD16, including many pointer driven programs such as a pointer driven Quill \_DOC file viewer, ScratchQL which offers drag and drop text transfer between most QL programs, and Doctor, a pointer driven utility for viewing and editing files

of any type. Also included are an updated U.K. STD telephone codes list, and Smash, a utility to compress executable files while keeping them executable. Qubbesoft were evaluating a ZIP disk of some 2,800 QL screens clipart and viewer utilities which may be available soon on EZ-135 and ZIP cartridges.

QBranch had a substantial presence at the show, offering their familiar wide range of products, probably now the largest single retailer of QL software. What they refer to as "the last ever batch of Super Gold Cards" have now been manufactured - last ever because a chip used on the board is now obsolete. This batch of Super Gold Cards does not have the 5 volt socket and can optionally be purchased ready for use with the Q-Plane from Qubbesoft P/D for assembly into a tower case, for example, although units converted in this way will not plug into a standard QL. Roy Wood and Steve Hall also displayed their Braquet plastic mounting adaptor for Aurora users to attach their Aurora cards to the standard mounting holes found in PC cases without having to redrill substantial parts of the cases. Keith Mitchell helped them demonstrate a batch of second user colour SVGA monitors being sold for just 65 pounds. These had been slightly modified for use with the Aurora to ensure they managed to display all available resolutions.

The new pointer driven accounts program, Q-Count, written by John Miller of the Bristol area sub-group, was available. Roy Wood was in full accord with those who say that pointer driven programs are the only way forward for QL software. It is becoming quite clear now that as software authors become more and more familiar with the pointer environment, most of the software being produced looks likely to be pointer driven, even PD/Freeware programs, so now is the right time to consider making the investment (time and money) to equip your system with pointer environment and to try to spend enough time to ensure you learn the system thoroughly as a user, even if you have no intentions of writing your own applications for it. As if to emphasise their commitment in this direction, QBranch have been supplying a free supplement with QPAC2 which gives a useful first introduction to the system by explaining how QBranch set up and use their systems using pointer environment. Go to a QL show these days and you'll find that most QL systems on display there seem to run this system, or at least have it on board. Of course, nobody is saying you have to convert to a system you think you may not be able to master, but most people who spend some time learning pointer environment and QPAC2 end up asking themselves why they didn't move to this environ-

ment on their systems sooner.

Since Quo Vadis Design decided to stop retailing QL software and become a software house instead, QBranch took the bull by the horns and took on board programs like Barry Ansell's Page Designer 3 and QIndex, along with most of the programs from Italian software house, Ergon Development. They also have v2.11 of Line Design from Belgian software house PROGS, now converted to running under Prowess, the extended window manager software used as the basis of all new PROGS programs. Prowess itself is also available separately from QBranch.

Bill Richardson of W. N. Richardson & Co. had on display a wide range of Spectrum, Z88 and QL bits and pieces. If you fancy obtaining a spare QL, or another QL circuit board to build into a cased system, or for the spare parts value, Bill can supply them for various prices dependent on condition. Also on this stand were the by now hard to obtain ED drives and diskettes. The drives on display were uncased Sony or Mitsubishi types, ideal if you wished to add them to a cased system for example. Spare parts are also available from this company, and they are now one of the few suppliers of those ubiquitous little serial to parallel converter printer leads, now manufactured by TF Services, who can also supply them in a version with a 9-pin D connector, suitable for use with the Aurora PC-style serial port connectors for example.

The TF Services stand was as usual dominated by the LEGO robot, being controlled by an interface from a Minerva Mk2 I2C bus. The latest release from TF Services was SuperHermes Lite, a cut-down and much cheaper version of SuperHermes. The Lite version offers the general Hermes improvements such as to the sound and serial ports, along with the keyboard interface from SuperHermes, though without the additional serial ports and a few other features. QL Today hopes to review this new product soon.

As usual there was a constant stream of visitors to the Jochen Merz Software stand, keen to update their software (buy some faster disk drives Jochen, PLEASE!) and to see demos of QPC, SMSQ/E and to subscribe to QL Today of course. An interesting development I noticed was that a number of ex-QL users who have migrated to using PCs (shame on them!) for their work, for example, were coming along having heard about the show via the Internet or local radio and heard that a software based QL emulator was available from JMS, enquired about the possibility of getting their old QL software running on their PC. Many had liked using the QL software, but the hardware had perhaps proved inadequate for their requirements, or they simply needed the

ability to use PC software. But it was obvious from their comments that they were not completely happy with these computers, that if they found themselves able to use their QL software on the PC without having to have two computers around their house, they would quite willingly do so once they knew what the emulator was about. Jochen Merz says that the fact he's picked up several QPC sales from ex-QL users who found information about QPC on the World Wide Web was behind his decision to start his own Web site. In a way it's quite funny to find Jochen eating his own words in this respect, as he's never been a great fan of email or the internet in general. Certainly, if a Web presence leads to him being able to sell QPC to many ex-QL users and thus bring them back into the fold, this is a development to be welcomed.

Jochen also demonstrated the latest version of his QD editor, now offering word wrap, margins etc, which while they do not make it a full blown word processor, certainly increase its usefulness, as word wrap means you can write simple letters etc on it but don't expect full text reformatting etc as you'd get in Quill or other QL word processors.

QMenu is now at version 7.03 and offers many new facilities added at the request of programmers like myself. For the BASIC programmer, most of the new facilities are implemented as optional extra parameters to the BASIC extensions, thus maintaining compatibility with older versions as long as you don't use the new facilities. Programmers who make use of these new facilities should make it clear in the documentation which version of menu\_rext (the extensions file from QMenu) their program requires. Since version 7 of QMenu is fairly new, many users will still have older versions on their system and the need to pay Jochen Merz a royalty on distribution of menu\_rext with a program means that it doesn't get included with PD/Freeware programs, so if such a program fails with a 'bad parameter' error on accessing a QMenu extension, this may be one reason, namely that their version of menu\_rext is too old. New features include the Extended Button, default directory option in DIR\_SELECT\$, ability to select files by type number rather than just extension in FILE\_SELECT\$, improved multi-line request options in READ\_STRING\$, sprite address options (rather than text) for buttons, and window colour enquiry for GET\_DEFAULT\$.

QPC was at version 1.20, including SMSQ/E version 2.84. There were quite a few colour notebook PCs running QPC at the show, demonstrating that the dream of a portable QL is in one way at least a reality. Marcel Kilgus, author of QPC, was also in attendance, answering

questions from users and making notes of new features requested. He was asked if he had any plans to produce a version which ran fully under Windows, rather than just in a DOS window as at present. Marcel replied that this would involve a huge amount of work, and was unlikely in the short term at least, although he would be looking at the possibility.

Geoff Wicks of Just Words was at the show, and is likely to be a more frequent visitor to British shows now that he resides in England. He had his range of software on display, and said that many of his customers had bought the special offer packs of his software, rather than just buying the one program. He also confirmed that he planned to look at producing pointer driven versions of his programs, where he believed that there would be a demand for pointer versions of individual applications.

Quanta's stand was staffed by the new Secretary, Roy Brereton, known to members as a past chairman and software head librarian. Roy now also doubles up as head software librarian, since the decision was taken to reduce the number of librarians to save on costs in this cost-conscious age, and also because demand for use of the library has fallen recently. Quanta are also looking at the possibility of moving to bi-monthly publication, or changing to A4 size newsletter, like QL Today. Roy was at pains to point out that the cost of membership at 14 pounds has not changed since the 1980s, so it was understandable that because of increased costs, options such as increasing the membership fee or steps to reduce costs had to be considered, and new chairman Robin Barker is known to be keen to tackle these issues and to sort them out as quickly as possible. Hopefully, if Quanta moves to bi-monthly newsletter publication, we can ensure that QL Today and Quanta appear on alternate months to give QL users who subscribe to both (I hope you do!) something to read every month. New editor Colin Baskett was in attendance during the first half of the show, getting to know people and generally talking about his plans for the newsletter.

Prolific QL software author and comms specialist Jonathan Hudson was in attendance, urging Dave Walker to produce a new version of C68 to fix a major bug Jonathan had found while attempting to compile one of his own programs. Jonathan is a major user of C68, and Dave Walker duly obliged, producing V4.22F of C68, which removed the problem Jonathan had encountered.

Dave Walker, meantime, was showing preliminary versions of his RLL (Runtime Link Libraries) system. I understand this is a facility many QL

software authors have requested, and Dave has finally managed to write this system. The first part is the RLM (Runtime Library Manager) system.

Albin Hessler had also travelled from Germany to be at the show, and he gave demonstrations of his Easyptr and CueShell systems. Easyptr now sports extensions to enable use of the `wm.rprt` timeout functions in recent window manager versions. A new BASIC extension, MCALLT, supports this facility, and EM\_RPTRT in the C and Assembler support packages. Easyptr extension files are now at version 3.5. Albin is also working on a new way to move and resize windows on the QL, by dragging and dropping the window title bars to move, and by dragging the outline border to resize the windows, reflecting Albin's background and experience in teaching people to use PCs and Windows generally. This is not yet part of the Easyptr package, but one day perhaps... As Albin now does his QL work exclusively on SMSQ/E, he said his future QL software products would probably only be guaranteed to work on SMSQ/E systems, further evidence that pointer environment is consolidating its hold on the QL software scene. CueShell is now at version 2.14, containing improvements in the handling of the SERNET S-device over networks.

I hope you can see from my show report just how much happens at these shows, and how much information you can pick up by attending, so keep in touch via the QL Today show calendar page and pop along to the next show in your area.



## Are you sitting comfortably? (Part 1)

Geoff Wicks

There are numerous laws and regulations governing the use of computers, but few apply to QL users. There are not many businesses with employees who use a QL, and most of us are using our QLs at home. Although many rules and regulations do not concern us, we should pay attention to them. They were not made to pester employers, but for the health and welfare of the computer user. Most QL users are the wrong side of middle age as far as our health is concerned and, for this reason alone, we should be looking after ourselves.

This short series of articles is based on health and safety information provided by the Dutch



(EEC) W.N. Richardson & Co.

Telephone & Fax: 01494-871319

Car phone: 0850-597650

**SALE  
PRICES!**

6 Ravensmead  
Chalfont St Peter  
Buckinghamshire  
SL9 0NB

## SINCLAIR QL / Z88

★ BRAND NEW QL CIRCUIT BOARDS - LATEST ISSUE - JS ROMS - £25  
NEW KEYBOARDS AND BASES - £15 UNPOPULATED BOARDS - £10  
KEYBOARDS INCLUDE MEMBRANES - BOARDS ARE COMPLETE ★

LOWER PRICES FLOPPY DISK DRIVES FOR QL AND PC  
(NB: 4Mb drives need Gold Card or Super Gold Card)

CAPACITY	2Mb DS DD & HD	4Mb DS EHD
SINGLE DRIVES	£75	£100
TWIN DRIVES	£110	£150
BARE DRIVES	£28	£65

IBM E.D.  
DISKS IN  
PLASTIC  
BOX 10 for  
£28 ONLY

## MONITORS

AURORA SVGA Monitors as new  
£85 + £12 UK  
Post + £20 EEC post.

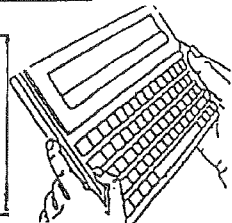
New mono's green  
or amber - £50

## MICRODRIVE CARTRIDGES

4 new cartridges in a wallet	£10	QL Psion Software;	
20 new cartridges in five wallets	£40	V2.35 Quill, Abacus, Archive, Easel in wallet	£18
8 program cartridges in two wallets	£15	Separate programs	£10
ZX MICRODRIVES & QL	£15	Interface 1 FOR SPECTRUM	£20

## SPARES

QL Printer Centronics Interface	£29	ZX-8302	£3
QL Membranes (with instructions)	£13	ZX-8049 (IPC)	£3
New top and bottom cases	£10	MC1377	£3
ZX-8301	£12	For other items please enquire	



## Z88 - THE IDEAL PORTABLE COMPANION FOR THE QL

File transfer kits available for the PC, QL, Mac and other computers

Includes built-in word processor, spreadsheet, database, BASIC, calculator, clock, alarm, calendar and VT52 terminal emulation. Uses four AA alkaline batteries (approx. 20 hours) or 220v a.c. power supply (included).

QL & PC and other computer users will find the Cambridge Z88 especially useful for work away from the desktop. With transfer programs data can be safely exchanged with other computers.

### PRICES:

Z88 version 3 - £95, Version 4 (OZ4) - £120

CENTRONICS PRINTER INTERFACES - £39

NEW RUBBER KEYBOARDS - £18

TRANSFER KITS: QL-£12 PC & MAC LINK - £25

RECONDITIONED Z88's - £50 with 90 day warranty.

RAMS and EPROMS - from £18

ERASERS - £29

220v Power supplies - £10

SEND FOR COMPLETE PRICES INCLUDING; RAMS, EPROMS, CENTRONICS INTERFACE, ERASERS.

POSTAGE: UK £5, EEC £15, USA £20, OTHER COUNTRIES £30.

ALL PRODUCT IS WARRANTED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING AGREED, AND IF THE ITEM IS OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE, PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

Trade Unions. They will cover the time we spend in front of our computer, the placing and furnishing of our work area, ease of software use and computer displays. They will be non-technical articles with practical hints that we can all follow.

## WORKING HOURS

There is a general rule that you should take a break of at least a quarter of an hour after each 2 hours in front of your computer. The quarter hours should not be saved up for longer breaks. This is a simplistic rule, and it is better to look at the specific task you are doing in determining the frequency and length of pauses.

**DATA INPUT** is often a repetitive, perhaps boring, task with little creative work. There is extensive use of the keyboard, and possibly the numeric pad. The eyes are moving constantly between keyboard, screen and document. Frequent pauses are necessary.

**DATA OUTPUT** involves long periods looking at the screen and good software is necessary to facilitate the task. Often it will be combined with other tasks, and there will be natural opportunities for pauses and creativity. If not, pauses should be created as it will be difficult to remain alert to the task.

**DIALOGUE** combines data input and data output and occurs in situations where a data base is searched. The software should facilitate this task. It involves both screen and keyboard work. Often there will be natural pauses and maybe opportunity for creative work. If not, pauses have to be planned.

**WORD PROCESSING** is a variable activity that can range from copy typing to creative writing. Mostly it will involve eye movements between screen, document and keyboard, and the ability to call up other programs such as a data base, thesaurus or spell checker. Good software is essential. The more routine the typing, the more necessary it is to plan proper pauses.

**PROGRAMMING** is usually an individual activity with a high level of creativity. It is screen intensive, but will often provide natural pauses as information sources are consulted. The high creativity of the task often means there is an inability to recognise that the time has come for a pause.

**GRAPHICS** and **CAD** also have a high level of creativity, and are probably the most screen intensive of all computer tasks. A good quality, well placed monitor is essential. Again creativity often means a reluctance to take pauses.

**GAMES** can be both screen and keyboard intensive. Pauses? What pauses? Don't interrupt me when I am just about to advance to the level of the seventh heaven and enter the Hall of Fame with my highest score yet!

You can use this list to determine the strains for you. What is your main use of your QL? Is it a use with natural pauses, or will you have to discipline yourself to take these? Is it keyboard intensive or screen intensive? Do you relax after a long computing session with a game, and then remain longer in front of your screen than you intended?



## QL VERSUS PC

QL users have an advantage over their PC colleagues. We have far more say over what appears on our screens. The ease of multitasking on the QL coupled with the flexibility of programs like QPAC 2 gives us many choices.

You can have one program at a time displayed on the screen, or several with overlapping windows. High resolution screens are going to give us more choices, but we must be critical in our use of these.

Does a high resolution screen aid or hinder my task? It is unfortunate that none of the QL word processors was written for high resolution screens. The display size of Text87 can be adjusted, but this does not alter the size of the displayed letter, which is the important thing. On the other hand a program like Line Design is much easier to use in higher resolutions. We may have to compromise in our use of high resolution screens.

The second advantage we have over PC users is our software. Does the software make our job easier or not? Is the display cluttered up with irrelevant information? Are the colours correct? Is there a border? As a QL-user we usually know our software supplier, and often the author, personally. We should not be afraid to talk to these people about our experiences and problems.

## HANDS, NECK and EYES

I shall be devoting a later article to monitors and software, but now a look at keyboard intensive tasks. There are computer tasks where you are constantly looking between screen, document and keyboard. There are frequent neck movements and constant refocusing of the eyes, both of which are tiring.

Now try an experiment. Load your word processor and type in a few sentences. Stop typing, but keep your hands in their normal typing position. Look at them. Are your wrists bent? If they are, then your typing position is wrong. At your next QL workshop watch other people typing. I think you will find the two finger merchants all have bent wrists, whereas the touch typists do not. If you do lot of word processing, it is worth the effort to learn touch typing. It is less tiring on the hands, and, as you will be looking less frequently at the keyboard, less strain on the eyes and the neck.

I could never discipline myself to learn touch typing on a typewriter. The exercises were repetitive and boring, my hands began to ache and I lacked perseverance. Learning touch typing on a computer is more fun. A computer keyboard requires less pressure than a typewriter, and the typing tutors are like games. You are given an instant score, which encourages you to persevere in making fewer errors and increasing your speed. There is constant encouragement to make the progress to move onto the next level.

The one I used, which I can highly recommend, was called "Type Right" by "Thimble Soft". I bought it from the old QL World Microdrive Exchange, so do not know whether it is still available or now in the public domain. Perhaps someone could do research into QL typing tutors and write a review of these.

Next time I want to look at the organisation and furnishing of your work area. For this you will need that most essential of computer peripherals....a high resolution tape measure!

■

## Getting into the States

Roy Wood

Back in the seventies, when I was an aspiring rock'n'roll muso I often day-dreamed about the possibility of 'breaking the States' in terms of achieving a Beatles - like fame across that continent and having vast riches, stretch limos, and bevies of underclad young ladies at my beck and call. Later, when I managed to download the 2 bytes of maturity that I have managed to survive on for the rest of my life, I abandoned the idea of being up on the stage in front of the lights and was happier harnessing the loud noises that came out of the sound systems. The idea of visiting the U.S. was still a bit of a dream and, since I travelled for a living, so to speak, I was adverse to the concept of travelling for fun. Unfortunately for this ideal there seemed to several gremlins in the way of my making the journey. I'd be working for an artist who would say 'After we've done the European tour we are off to America' and I would start to plan things only to find that one of the many pitfalls of the entertainment business had opened up and either swallowed the budget ('I'm sorry, we can only take the wardrobe'), or the artists themselves so I never ever made it across the pond. It began to be a bit of a Nemesis. Early this year Jochen asked if I was going to the American show and, mindful of the horrors that had accompanied earlier attempts at transatlantic travel I declined. Jochen and Stuart then started a concentrated campaign of emails, faxes and phone calls until I caved in and booked a ticket. Q Branch is not a rich company so I was very worried about the cost of the journey but I stuffed a suitcase with QXLs, recycled Gold Cards, a couple of Super Gold Cards pre-ordered by customers and software and set off for the Airport. I met Tony Firshman in the airport lobby and he gave me a packet of SuperHermes, cables and the like which I added to my suitcase. 'Did you pack this case yourself, sir?' was the first query 'Er, well, er' 'Are you carrying anything for anyone else?' ('lie!', I thought). I had only booked the flight on the Monday to fly out on Friday so the travel agent said that the tickets would be at the departure gate. They weren't - they were at the travel agents (The Gremlins, The Gremlins!!). Finally, after numerous phone calls, we managed to sort out the problems and I was airborne, so far so good. When I landed, after a smooth and

comfortable flight, I realised that when the immigration official asked me where I was staying in the U.S. I wouldn't know. Stuart had booked the hotel and I was meeting Jochen and Jim Hunkins at the airport so we could share a car. I bluffed my way around that problem and even managed to meet up with Jochen and Jim as arranged and very soon we were driving out of Washington en route for Bedford PA. We arrived at the Hotel in the early evening and immediately rushed down to the venue to have a meal and meet the members of NESQLUG who had organised the show. This was a great evening and I got to meet a lot of people who I had only spoken to on the telephone or communicated with by email or post. They are such a friendly bunch of people, the American QLers, and we sat around chatting for some time. I am not sure if it was the jet lag or the copious amount of Californian red wine but Stuart and I had a great deal of difficulty in walking back to the motel. It didn't help that the tarmac the road was made of just petered out unevenly at the sides.

The next day we were up bright and early and breakfasting at the diner across the road from the Motel before driving down to the show to set up. Many of the U.S. traders had set up the night before and Jochen and I settled into the corner to set out our wares. Jochen had arrived with all of the manuals as loose leaves in order to defeat the customs so he had to put it all together in the Motel the night before. He was a trifle more bleary eyed than I was and he had not been drinking. Sales were brisk and pretty soon I had sold all of the ProWesS programs I had brought along with me and was taking orders for more to be sent on. The Super Gold Cards and QXLs that I had went quickly as well. One novel touch here was that the talks took place in the same room, very close to my stand, so I was able to actually hear some of the discussions that went on. Most enjoyable. Another new feature was that half way through the day we broke off for lunch and everyone vacated the room for a meal provided by the show's organisers. This gave me a chance to sit down and talk to Don Walterman who had given me a lot of advice when I was struggling with my old EPSON 286 laptop. They also had a raffle at this point. Each of the traders had donated a piece of software or hardware and the winners were able to take their pick of the items.

Herb Schaaf gave us a very interesting display of the animated Escher drawings he had put together, an amazing piece of programming that had me feeling dizzy after looking at them for a while [**Herb started writing; more in the next issue - Editor**]. At the end of the day I had very little left to take home and a big pile of orders to be fulfilled. We packed up our stalls and headed back to the Motel for a quick shower before yet another meal (and more Californian Red).

It was the next morning when I realised that I had to fly back to England the next day and it suddenly dawned on me that I had travelled all this way and would be seeing very little of America before I went home but Stuart Honeyball helped out by navigating us back to Washington Airport. He took control of the directions and managed to send us on a detour that took in some wonderful countryside and eventually put us onto the wrong side of the Potomac River ("There seems to be a blue wiggly thing in the way"). He kept saying that we were on the right highway but didn't realise we were in the wrong state ('Get confused on route sixty six').

Jim and Jochen were getting a bit worried about catching their plane by now (ours was not due for hours) but we finally pulled into the Airport on time and they zoomed off for a short holiday on a rollercoaster (?) while Stuart and I wandered round one of the most empty airports I have ever been in. We did think about travelling into Washington for a sight seeing tour but found out that, by the time we got there, we would have to turn around and come back so we slouched around the airport until it was time to get on a plane. Of course the gremlins struck again and I found that the plane was overbooked and they were asking for volunteers to go on the next one (and offering a \$400 reward). I took it and that is \$400 towards my next U.S. trip because I am definitely coming back. My thanks go out to all who arranged and attended the show. See you next year.

■

## Bugs 'n Fixes

**Question:** I can't play audio CDs in QPC using CDPLAYER.bas. My DOS config.sys has a reference to "Mscd000" as required for the CD\_INIT command. I'm using Windows 95.

**Answer:** Since the DOS window used by

QPC had its own autoexec.bat and config.sys files, which were different to those used by the MSDOS Prompt DOS window, it turned out that the DOS CD driver was not being loaded by this alternate config.sys. We went to the DOS shell of QPC (CTRL ALT SHIFT Scroll-lock), typed in the command TYPE C:\config.sys to check this was the case, then entered the command EDIT c:\config.sys, added the line from the main config.sys which loaded the CD-ROM device driver (this may vary from system to system):

```
Device=C:\Sony\Atapi_Cd.sys /D:Mscd000 /I:0 /G
```

This was then saved, we quit from QPC and then restarted it and from then on the CD player worked fine.

**Question:** I get a "not found" message when I try to start QPC, even when I try to start from floppy disk!

**Answer:** You have probably moved the file QPC.EXE out of the directory where all the other QPC-files (especially SMSQE.BIN) are located. Instead of moving the QPC.EXE file, create links to this file which you can freely move around, place them on the desktop or put them into the START menu.

If you try to boot from disk after installation,

then if your prompt is something like

```
C: >
```

and you try something like

```
A:QPC.EXE <RETURN>
```

then you will get the "not found" message, because QPC tries to find the other required files on the "current" drive, which is "C:". Use instead

```
A: <RETURN>
```

```
QPC.EXE <RETURN>
```

which will work, because A: makes the floppy disk the current drive, so the other files will be found there.

**Roy Wood replied to Anthony Trice's questions. Sorry that it could not make it into the last issue, which was absolutely full:**

First I need to know what versions of SMSQ/E and ProWesS he has. Earlier versions of ProWesS had serious problems on installation. One of these was that when told to write a boot file it promptly deleted anything on the destination device with boot in its name. Those people who had more than one boot file for different jobs (such as boot\_Text87, boot\_LINEdesign etc) quickly found out the value of keeping backups. The second problem was that when you tell it to configure the program on installation it only

## WE SUPPORT SINCLAIR

QBOX USA

COMPUTER BULLETIN BOARD SERVICE  
(810)254-9878

**We support *all* SINCLAIR COMPUTERS  
(QL, SPECTRUM, ZX81, Z88, Thor, QXL)**

- Now in our 3rd year on-line round the clock since October, 1993
- Full message area and File Download areas
- We carry all popular SINCLAIR message areas from Europe
- Calls from 14.4k--300 baud are welcome
- QBOX - USA runs on a SINCLAIR QL with Super Gold Card, Hermes, QUBIDE, 200MB drive, USR sportster 14.4 modem

**→ NO FEES CALL US**

configured the program on the floppy not the hard disk you were installing it onto. Both these problems are now solved. Phil Jones has reported that the program will not install onto a JS ROM because of a fault in the JS ROM's COPY command. SMSQ/E, however, should be a different matter. It is written to make the most use of SMSQ/E although you do need a version higher than 2.79 to get the best results (screen events). Using the Aurora patch takes out the screen events but this should be solved when Tony Tebby finally gets around to making a new version of SMSQ/E that works on the Aurora [done, see JMS-advert]. I have installed ProWesS onto several systems with SMSQ/E present and had no problems at all. Just place the disk in flp1\_ and run the boot file. There are traps in the boot file that will not load the Pointer Environment if SMSQ/E is present. I really need to know more about what he is doing and what problems he is getting before I can make an assessment. Once ProWesS is installed onto the hard disk you have to decide how you want to put it onto your own boot. I have used the following lines:

```
LRESPR win1_pws_ext_dll_rext
LRESPR win1_pws_ext_syslib_rext
LRESPR win1_pws_ext_global_rext
LRESPR win1_pws_ext_PWbasic_rext
```

you also need the QL:IB\_run resident for some of the utility programs. At the end of the boot file you should also add the line:

```
EX win1_pws_prg_loader;"win1_pws_startup"
```

The startup file will load a few more extensions and then start the pws\_mine\_personal\_ldr file which starts the buttons. This is a plain text file that can be edited by the user to add extra buttons to the frame. This is done by adding a line calling another ldr. For example to add the QPAC 2 programs in one button add the line:

```
Cbutton -name 'QPAC2' QPAC2_ldr
```

This will call another file which should be in the win1\_pws\_mine\_ sub-directory and should look like this

```
mbutton QPAC2_mbt -name QPAC2
```

There should also be a file in the same sub-directory and consists of all the names of the programs loaded resident when QPAC 2 is LRESPR'd - like this:

```
'Channels'
'Exec'
'Jobs'
'Hotkeys'
'Hotjobs'
'Pick'
'Wake'
'Rjob'
```

```
'Sysdef'
'Things'
'Files'
```

Next time you run the boot file you should get a button called 'QPAC2' which will present you with a menu as above when clicked on. Click on any of the items in the menu and the appropriate program will be started. Finally - I know this is a bit long winded pick the bones out for yourself - I will give a quick rundown on how to install programs into ProWesS. When ProWesS is installed into the system you should have an button called 'Utilities'. In this menu there are two items: 'Start program in flp1\_' and 'Install Software'. The first item will start any piece of ProWesS based software in flp1\_. The second item is used to install the new piece of ProWesS software to the hard disk. Click on this item and a box will come up checking that this is what you want to do. Indicate 'do' and you get the option to:

```
'Install on a separate button'
'Install on an existing button'
'Install on a new menu button'
'Update installation'
'Add documentation to bookmarks file'
```

The first option will install the program onto the hard disk and then add a button to the button frame that will start the program directly in a similar way to the normal QPAC 2 buttons. The second option will install the program so that its name appears as a menu item in one of the existing buttons'. The third option will install a new menu button with the program's name and the program as an option in the menu. If you choose this option you can then add related programs to this menu and have them all available on the one button. The next option does what it says. The final option adds the location of the documentation to the bookmarks file for the ProWesS Reader so that you can click on the 'Bookmarks' item in the Reader and look up the manual. Both of these items will only work with ProWesS applications because they look for specific files on flp1\_ but, as you can see from the QPAC 2 button stuff you can install anything into a ProWesS button. If the file is not LRESPR'd then all you have to do is write the line (for example to load and start QTPI):

```
QTPI -name 'QTPI' -path win1_qtpi_
```

(my file is win1\_QTPI\_QTPI) into the 'Applications\_mbt' file and - hey presto - there it is ready to use. You do not even have to reboot the machine. Since these files are all plain text files it is very easy to do this kind of thing. I hope this is useful.

■

# Letter Box

*This months Letter Box also deals with our competition entries. We printed the best suggestions and ask you, the readers, to vote for their favourite term.*

*[Sorry, this letter did not make it into issue 2 - it has been cut from its original length] P.H. Tanner, Glasgow, Scotland writes:*

I like your new font. The second biggest problem with IQLR was its illegibility. The ratio of ink to paper was far too high. This was made obvious by the ads which appeared as completely legible islands floating in an impenetrable sea of editorial wodge. I found it impossible to read, and never renewed my first sub. You have my continuing support.

Another bull point for QL Today and its editor. Proportional fonts are death on program listings. The majority of programmers take great care on the layout of their work. But source editors assume that the full "em" will be used for spaces, while DTP programs default to the proportional "en". Which can result in chaos. *[We took a conscious decision to use a non-proportional font for listings in QL Today, as we were aware of the problems which could arise when using proportional fonts - Editor]*

A minor matter. On page 17 of issue 1 the writer states that "...on the PC... CAPS LOCK forces the computer to read the characters above the number keys (0-9)..." I believe this only to be a feature of the Amstrads. And perhaps not all of them. I have never met another breed of PC that did this. But I may be wrong. I often am. *[This is a "feature" which seems to work on most PCs. It also does on mine. Maybe somebody should point out to Mircrosoft, that numbers are not capitals! - Editor]*

There is a need for an information clearing house on QXL matters. When I purchased mine nearly 4 years ago I raised the question of its being fitted with the full 68040 with on chip co-processor. I was told by Miracle that this was not possible. In the interim I have cobbled the basic QDOS/SMSQ arithmetic to make it more suitable to my needs, and rubbed along as best I could. This year, it has become apparent that not only was that mod always possible, but that people have been making it successfully. And now we have this article by S. Goodwin. A lot has been happening on the inside track of which no word has leaked out to the ordinary punters on the terracing. (Since this letter was started, the May QUANTA has come to hand, with yet more on the 68040. Where was all this activity previously recorded?)

Right now, I am rejigging my programs to take advantage of an enhanced QXL. But I have lost that 4 years which is a rather greater proportion of the active time that I have left than it would be for the majority of your readers. *[Go for it, and let us know about your activities too! - Editor]*

I have found another SMSQ funny while integrating the GWASS60 assembler with my existing conditioned reflexes through the system. COPY\_O is proving most insecure when used to overwrite floppy files. Although all the new blocks are correctly written into their sectors, as I can verify by inspecting the disc with Norton Utility, the FAT is not always fully updated. I am finding that a DELETE followed by a plain COPY is much safer. Which makes my usage of the extra QXLWIN on my virtual disc, giving me a volatile win2\_, even more useful. *[see Some QXL & QL Tools Notes, Vol 1, issue 6 - Editor]*. I can treat it as a scratch substitute for the floppy, doing a final WCOPY at the end of the session (why should WCOPY work, and COPY\_O not?)

*And just another letter from Mr. Tanner:*

Liked the July/August issue muchly. Especially Mr. Al Boehm. Made me think - ouch! I thought that I had left that unaccustomed exercise behind me.

What we need is to resurrect Mr. Bryan Johnston - the former "Troubleshooter". Most useful when there were problems with traders. He was also good at relaying the gossip from within the inner circle.

"QLers". Careful now.

I am a computer user who happens to find QL-type machinery most suitable to some, but not all, of his purposes. If you lot were to retire to a self-imposed ghetto, and pull up the drawbridge, not everyone may follow.

QL Today is great. Please keep it up.

*John Miller writes:*

Dear Dilwyn,

I thought that I'd enter your competition to rename "QL users", as it's something that I've been thinking about without success. However, with the incentive of a prize, here goes!

My first attempt is "SQL users" - pronounced "sequel users", on the grounds that we are mostly using a sequel to the QL. However, this probably falls foul of the likelihood that there is probably already a SQL Users Group!

My partner, Debby, suggests that we should be called "QLITS" - "QL In There Somewhere"!

But I think I'd like to suggest that we take a leaf out of Arthur C. Clarke's "2001 - A Space Odyssey", where the computer, HAL, was named by taking the letters before IBM. Most of us are decline to be seen as PC users, so how about taking the next letters, and being QD users? In this case, QD stands for "QL Development"!

Keep up the excellent work on 'QL Today' - I'll write another article for it sometime. *[now that promise does earn Brownie points - Editor]*

**K.J. Brickwood, London, writes:**

#### **PREAMBLE**

These days, we can run our favourite programs on such a wide variety of equipment it would be hard to find a 16 or 32 bit machine which isn't QL-Compatible. Devising a hardware-based term that can be applied to anything from an Atari ST to a Silicon Graphics Workstation is virtually impossible.

On the other hand, everyone in our community uses QDOS and/or a QDOS compatible operating system. Therefore, it would make sense to use a name which represents this underlying similarity, but reflects the differences.

#### **A NEW NEW COLLECTIVE NOUN FOR A NEW MILLENIUM**

I propose that we should call ourselves: QSA Users.

#### **WHY ?**

Q = QDOS, Q-EMULATOR, QL, QLAY, UQLX, QPC, QXL, QXL II

S = SMS, SMS2, SMSQ, SMSQ/E

A = ATARI mono, ATARI colour/QVME, QL & AURORA, ARGOS/ADJUSTED QDOS (Thor), AMIGA QDOS, ATARI QL EMULATOR CARDS

#### **WHAT ABOUT MINERVA ?**

Minerva is essentially an enhanced and debugged QDOS implementation - the operating system the QL could/should have had if Sinclair Research hadn't made such a pig's ear of the project.

**John Wakefield writes:**

Rising to the challenge of your competition, I started thinking along the lines of who has made the most significant contribution to the QL / SMSQ (E) / etc. scene over the years, and the name which came to mind, among other notables was Tony Tebby. I thought it would perhaps be appropriate to make an anagram of his name, and the best I could do was 'BYTE-Y-BONT'. This, as you must recognise, is a 'BIT' of a well known 'ADDRESS', but I cannot see the QL fraternity adopting it as a group name.

I next turned my anagrammatical attention to another esteemed personage, but 'HA-NOBELLY' is not how I remember him, and so I could only think of 'MIRAQLERS'. How bad can this get? *[If you've seen him eat during show dinners etc, you'd realise the word 'belly' might come into it somewhere, I only wish I could eat as much as him and remain his size! - Editor]*

I Quickly dismissed 'SMSQEERS' AND 'QDOS-SERS', and an anagram of a well respected magazine 'QLTOADYS' did not seem respectful enough. *[I'm well used to thinking of PC users as living in DOS-Houses, but I hadn't thought that I might live in a Q-DOS-House - Editor]*

Since we are all followers of 'The Cause' I considered 'SEQLers' (sort of sequel ers, which means following, does it not?), which could also stand for 'Sinclair's Everlasting QL'. Then, as we are banded together, 'QLIQUERS' (which also sounds like what we do with a mouse).

My final unworthy effort is 'QPACKERS', because every QL family user must surely be using that wonderful program QPAC2, Aren't they?

If any of my suggestions win, I can only assume that either the judge is a brother loony or the rest of the contestants are boring old pharmacists.

p.s. I am still working on an anagram of Jochen Merz.

p.p.s. Did I ever tell you that my grandfather was Welsh? That should be worth a few Brownie points. *[Probably worth a few Brownie points, but attempts to bribe or otherwise influence the editor's decision just cost you a few points! - Editor]*

**Robert H Klein, Germany, writes:**

Nice idea this competition. I have some answers to it.

The first term is (guess) "QLers." You have yourself written in the article of a "new collective term to describe QLers." So this seems to be the best word I can think of. It has some slight disadvantage though. One time at a QL show in Eindhoven someone wanted to buy a disk from me, and as it was a HD disk I wanted to go sure he could read it. So I asked him "Can your QL read HD disks?" (Yeah, bad English, I know...) He told me no. I tried to explain to him that he could do nothing with the disk if his QL could not read HD disks. Some time down the discussion it turns out he has a QXL where he wanted to use it. Well, I thought about some nice terms to describe this guy some time after. Look up "jerk" and "bonehead" in your dictionary. But in a narrower sense of the word he was right.

Another term that fits most QLers (plus a lot of other people) is "nerd," though I think that's not in discussion.

I don't think there can be one new catchy term that has no reference to at least either the QL or its operating system QDOS. As most emulators emulate first the QL's hardware and then run QDOS or a derivative of it there is plenty of room for invention. Something that shortens "QL/QDOS successor thingie" in a nice and convenient way. But I like "QLer" most.

Okay, enough on that, to the CIA world fact book reviewed some time ago.



# Steve Johnson

QL Public Domain, Freeware And Shareware.

CompuServe ID = 101325,2750

Internet Email [SteveJohnson2@compuserve.com](mailto:SteveJohnson2@compuserve.com)

36, Eldwick Street,  
Burnley,  
Lancashire,  
England.  
BB10 3DZ.



Tel / Fax +44 (0)1282 701767.

## SOME OF THE LATEST ADDITIONS

QTPI RELEASE 1.6.2.4.  
QFAX RELEASE 2.8.2. (3/4)  
QL HACKERS JOURNAL. (2)  
XCHANGE VERSION 3.90L. (2)  
G. WICKS DEMO DISK.  
POINTER MICROEMACS. (3)  
LINEDESIGN FONTS. (16+)  
LINEDESIGN CLIP-ART. (50+)  
MONOP. (MONOPOLY)  
TREK. (STAR TREK)  
SPECTATOR VERSION 1.85.  
LIB LIST. VERSION 1.1.  
SMALL "C" COMPILER.  
MARK KNIGHT UTILITIES DISK.  
"GOING ONLINE" TEXT FILES.  
SBASIC PE KIT VERSION 1.3.  
PROWESS DOCUMENTATION.  
D-I-Y TOOLKIT. (3)  
GEORGE GWILT'S  
ASSEMBLER & DISASSEMBLER.  
NET PEEK & FPSAVE.  
FARM CLIP ART. (3)  
MUSIC CLIP ART. (3)  
QTYP DICTIONARIES.  
Z88 SOURCEBOOK  
SCREEN DAZZLER.

THIS IS ONLY A SMALL SAMPLE OF THE 180+ PACKAGES ON  
OFFER.

THE NUMBER IN BRACKETS = THE NUMBER OF DISKS IN THE PACKAGE.  
NO. NUMBER = 1 DISK PACKAGE.

*I can also offer 75 disks of the  
best in assorted QL Public  
Domain, Freeware &  
Shareware.*

*I stock a full range of emulators  
for use on either IBM  
compatibles or the Amiga.  
Including the latest releases of  
QPC & QLAY  
The Sinclair QL emulators.*

*I can now offer over 250 disks of  
classic literature & assorted text  
files. Including the complete  
works of Shakespeare.*

*Free disks available!*

*Please see catalogue disk for  
full details.*

QL pre-copied disks are £1.25. Each, fully inclusive of media and return UK postage.

Please add 25p per disk for none UK orders.

For a catalogue disk which details over 500 disks of PD & Shareware & Text Files offered.

Send a QL formatted disk and return postage.

Payment by Cheque made payable to S. Johnson.

English Banknote Also Accepted. No Facilities for Credit Cards or Eurocheques.

*"Supporting The Sinclair QL Into The Next Millennium."*

01/09/97.

You can find a new version of the CIA world fact book at

<http://www.odci.gov/cia/publications/nsolo/wfb-96.txt.gz>

**P.Lake, England, writes:**

Quite a challenge to find something of a generic term that can encompass members of such a wide range of international growth both in software and hardware, taking the original concept so far beyond its initial form that it is hardly recognisable. The following suggestion is given, together with some of the combined implications:

QLIVERS (pronounced "Clivers")

- in memory of Sir, Genesis of the electronic concept.
  - QLIVers
  - Q livers (alive, not the organ!)
  - qliver (cliver that is, from clifer in mediaeval english, later to become 'clever'.
    - Of persons 'possessing skill or talent; able to use hand or brain readily and effectively.'
    - Of things 'Done or performed with adroitness, dexterity or skill, ingenious.'
- (Compact Ed. Oxford Engl. Dict. pp 494-495)

**G.T.Morris, England, writes:**

Here is my suggestion for the name or title of those computerised enthusiasts who continue to recognise the delights and joy of using that Sinclair concept called The Quantum Leap.

The name for us, the users, should be ... QUANTICS.

I am quite satisfied with my club journal QUANTA, may its life continue, and my magazine for QUANTICS INTERNATIONAL (there's a thought).

Why QUANTICS? You can guess at the obvious reason, but the dictionary says it all:

Rational integral homogeneous function of two or more variables.

Can we say that QUANTICS use an OS QUANTifier called a QUANTIC QDOS or a QUANTIC SMSQ?

It's the QUALITY of us QUANTICS that matter not the QUANTITY, although there are plans afoot to QUALIFY even that! Let's hope it succeeds.

**Now, readers, is it up to you to tell us what you prefer most. Write a postcard to Jochen Merz Software or Dilwyn Jones with the idea you preferred most. The winner with the most votes will get the Fontpack, and there will also be a price draw for the voters: every postcard has a chance to win one of three QL Today 1998 wall calendars. The deadline is 15th of October 1997.**

■

## Mailbox & eMail Shortcuts

Jochen Merz

If you read messages in BBS's or get eMail, you often find strange character combinations like :-) or IMHO or RTFM. The latter two are just abbreviations to save the writer time to type them in, because they are well known abbreviations in the "Online Scene". The first character combination is called "emoticon" or "Smiley".

When you write messages or comments and you criticise somebody, but you don't mean it too seriously, it may be a good idea to add an emoticon so as not to upset the person who reads your message.

For example: If you say to somebody "You can't be serious", then the real meaning of this sentence depends on the context and on your expressions. As the reader cannot see the expressions of your face while he reads your text, adding a smiley will help. Reading a smiley is quite easy: just turn your head slightly so that the left hand side of your monitor becomes top and the righthand side becomes bottom. If you have a notebook, turn this instead but don't try to turn your 20" monitor!! ***[[ wonder if Jochen actually tried this, hence the warning? - Dilwyn]***

Immediately you can see what I mean :-) the colon represents the eyes, the line is the nose and the bracket is a smiling mouth. You can add a wink by using a semicolon instead of the colon, for example. As far as I know, over 340 registered smileys exist, but here are some of the most commonly used:

- ;-) This was ironic.
- ;-> This was really sarcastic.
- :~D Really funny, the stuff you wrote.
- :~o I'm rather suprised about your comment.
- :~x This is top-secret and I keep my mouth shut.
- :~\* Kiss.
- :-( Disappointed.
- :-( ( Very disappointed.

If you wear glasses, replace the colon with an eight, e.g.

8-)

If you wear a toupe, use

{:-)

After a really diabolical comment, add

>:-)

Bonzo the clown

\*:o)

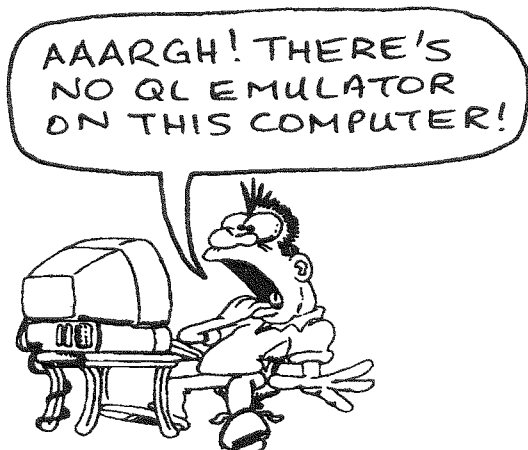
Don't spread too many smileys into your texts, otherwise the reader may not take them too seriously.

And now on to abbreviations: long words do cost money when you send them over the telephone line, so why not use an abbreviation which everybody knows instead. If you get used to them while you're writing offline, then it's even better when you use them while you're typing online, i.e. chat with the Sysop.

AFAICS As far as I can see  
AFAIK As far as I know  
ASAP As soon as possible  
BTW By the way  
EGBOK Everything's going to be OK  
FYA For your amusement  
FYEO For your eyes only  
FYI For your information  
IDU I don't understand  
IMHO In my humble opinion  
IOW In other words  
JFYI Just for your information  
NOM No offence meant  
OIC Oh, I see!  
OTOH On the other hand  
PITA Pain in the a\*se  
POV Point of view  
RTFAQ Read the Frequently Asked Questions  
RTFM Read the "fine" (=f\*\*\*ing - guess what it stands for) manual  
(e.g. Read The Flaming Manual)  
SIDU Sorry, I don't understand  
SISDU Sorry, I still don't understand  
TAFN That's all for now  
TTMS Type to me soon  
WDYM? What do you mean?  
WDYW? What do you want?

You probably can't remember them all at once, but try using them and you will very soon get used to using and reading them. CU (See you).

■



# JOHANN MERZ SOFTWARE

## No QL emulator on your PC? Get more, get

# QPC

With QPC, you can run most of the current QL-software on PC's. You need at least a 486, but QPC will run faster on a Pentium. 4MB RAM, EGA graphics card, MS-DOS 3.3 (or higher) or Win95. QPC can now easily be installed to be called from Win95 - the manual has been rewritten!

A double-mouseclick can turn your PC into a much better, QDOS-compatible system. Better, because with QPC you get Tony Tebby's new operating system SMSQ/E for QPC - it is included in the price!

Do not worry about any soldering, plug-in cards etc. - QPC is a software emulator, it does not need any extra hardware! This means, you can install it even on laptops!

QPC offers access to the serial ports (up to 57600 baud!), parallel port, harddisk and floppy disks. It can read and write QL and DOS floppy disks, so data exchange is easy. You can playback audio CDs even from within QPC. PS/2 and serial mice are supported.

The new display driver not only supports 512x256, 640x350, 640x480 and 800x600 pixel resolutions, but with the new VESA support also even higher ones (e.g. 1024x768).

QPC is not expensive: you get the emulator plus the operating system SMSQ/E for only **DM 249,-**. If you own SMSQ/E for another system you pay only **DM 199,-**.

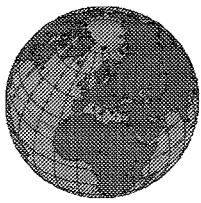
If you want to get the excellent CueShell Desktop program from Albin Hesser bundled with QPC, just add **DM 40,-**.

## Test QPC!

A demo version which will do everything the full version does (except writing to floppy and harddisk) is available for only DM 6,- including p&p (or send 3 international Reply Coupons).

## Version 1.30 exists!

10 - 20% faster than 1.20 or 1.21. Updates are free, get them via BBS or the usual way: send the disk, include 3 IRC's. See how good it is - review in current issue!



# The QL Show Agenda



**Saturday, 13th of Sept. 1997** East Midlands Quanta Workshop, Selston Parish Hall, Selston, Nottingham, England. The venue is close to the M1 (Jct. 27/28) and has easy parking on-site. Overnight accomodation a few miles away can be arranged. A good discount may be possible if there are enough staying, so please contact Graham Underwood 01909-531405 or Dennis Smith 01773-748740 to give them an idea how many rooms to book in advance.

**Sunday, 5th of October 1997** Byfleet Village Hall, Surrey, England. Contact Tony Gordon 01372-458180 or Ken Bain 01932-347432.

**Saturday, 8th of Nov. 1997** The Netherlands, Eindhoven, St. Joris College - same venue as always.

**Sunday, 30th of Nov. 1997** The Bristol Workshop - for details see below.

---

**2nd Sunday every Month** Quanta London Sub-Group meeting. St. Aloysius School Hall, Phoenix Road, NW1. Meeting starts at 13:00 and ends at 18:00. Contact Basil Lee 0181-789-1976 or basilee@mail.bogo.co.uk

The Bristol QL User Group is very proud to present once more for your delectation, the event in the QL calendar you have all been waiting for:

## **The Bristol Workshop**

being held at

**Somerset Hall, Portishead, Somerset, Nr. Bristol  
on Sunday, 30th November 1997 - 10.00am to 5.00pm**

All the features which have in the past marked this event as outstanding will be present:

- Separate traders' and users' areas
- Two separate product presentation areas
- Forums for special interest areas
- Famous Bristol Bring 'n Buy
- Help and Advice Clinic
- Theatre style lectures/teach-ins

***See you there!***

Come and see what all the others are talking about, and you too can become part of something special. The event is well signposted from Junction 19 of the M5, right in the centre of Portishead.

If you need more details, please contact your local QL user group or QL Today!  
You can help to turn QL shows into even more interesting events!

Also, if you (plan to) organise QL shows or regular local user group meetings,  
please let us know and we publish it here!